《从零开始学 Python》(第二版)

极客学院出版

[illegible]

Python [illegible] Guido van Rossum [illegible] 1989 [illegible] 1991 [illegible]Python [illegible] C/C++[illegible]

Python [illegible]Python [illegible]Python [illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] Python [illegible] Python [illegible] Python [illegible]

[illegible]

[illegible] Python [illegible] Python [illegible] Python[illegible] Python [illegible]

[illegible]

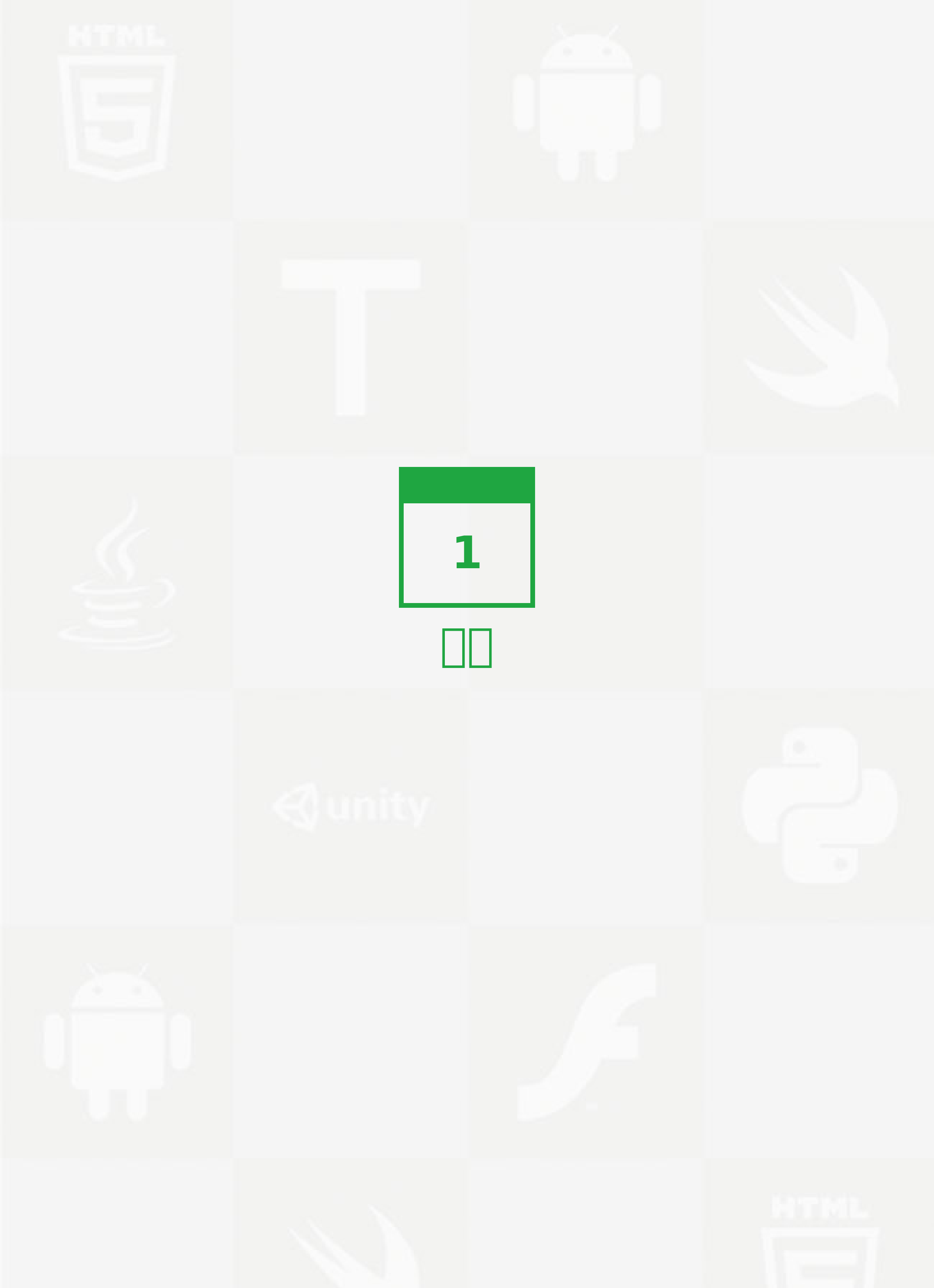
[illegible]“[illegible]”[illegible] Python[illegible]

[illegible]

[illegible]

[illegible]/[illegible]	[illegible]
Python	2.7.6

> [illegible]: [illegible]
> [illegible]: 2015 [illegible] 7 [illegible] 30 [illegible]



1

[illegible]

□□ Python □□□

□□□□[□□□□□ Python□□□□□□ href="https://github.com/qiwsir/ITArticles/blob/master/BasicPython/index.md)□□□□□□□□□ Python □□□——□□□□□□ Python □□□——□□□

□□□□□ Python

□□□□(before 2011)□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□

> “□□□□□□□”
> “□□□□□□□□□□□”
> “□□□□□□□”
> “□□ php□java□c# □□□□□□□□□”
> “□□□□□□□ pascal□□□□□□□□□□□”
> “□□□□□□□□□□□□□□□□□□□□□□□□□□□”

□□□□□□□□□□□□□□□□

2014 □□□□□□□□□□□□ Python□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□ QQ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□ Python □□□□□

1. □□ QQ □□□□□□□□□□□□□QQ □□Code Craft□26913719
2. □□□□□□□□□□□□□□□ Py□□□□http://weibo.com/qiwsir

3. [illegible] github.com [illegible] follow [illegible]qiwsir[illegible] https://github.com/qiwsir
4. [illegible]www.itdiffer.com

[illegible]2015 [illegible] QQ [illegible]26913719[illegible] Python [illegible] Python [illegible]

[illegible] Python [illegible] Pythoner [illegible]

[illegible] Python[illegible]

[illegible]

[illegible]

[illegible]Python [illegible]

[illegible]“[illegible]”[illegible]

[illegible] Python [illegible]

[illegible] C [illegible]

[illegible] Python[illegible]

[illegible] **Python**

Python [illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible]

[illegible] **Python**[illegible]Python [illegible]

[illegible]“[illegible]”[illegible]

[illegible]

“[illegible] Python [illegible]”——[illegible] Python [illegible]

[illegible]([illegible])[illegible]

RTH CAROLINA
HAWAII
ARIZONA
UTAH
ALABAMA
CALIFORNIA
TUCKY

> PHP [illegible]
>
> Ruby [illegible]
>
> Python [illegible] Ruby [illegible]“[illegible] Python [illegible]”[illegible] Python[illegible] Ruby[illegible]
>
> Java [illegible]“[illegible]”[illegible]
>
> C++ [illegible] Java [illegible] Java [illegible]“[illegible]”[illegible]“[illegible]”[illegible]
>
> C [illegible] C++ [illegible]
>
> Objective C C [illegible]

[illegible]

The Zen of Python

[illegible]Python [illegible]

> Beautiful is better than ugly.
>
> Explicit is better than implicit.
>
> Simple is better than complex.
>
> Complex is better than complicated.
>
> Flat is better than nested.
>
> Sparse is better than dense.
>
> Readability counts.
>
> Special cases aren't special enough to break the rules.
>
> Although practicality beats purity.
>
> Errors should never pass silently.
>
> Unless explicitly silenced.
>
> In the face of ambiguity, refuse the temptation to guess.
>
> There should be one-- and preferably only one --obvious way to do it.
>
> Although that way may not be obvious at first unless you're Dutch.

> Now is better than never.
>
> Although never is often better than *right* now.
>
> If the implementation is hard to explain, it's a bad idea.
>
> If the implementation is easy to explain, it may be a good idea.
>
> Namespaces are one honking great idea -- let's do more of those!

[illegible] **Guido van Rossum**

Guido van Rossum [illegible] Pythoner [illegible] Python [illegible]

> 1989 [illegible]·[illegible] ABC [illegible] Python [illegible]·[illegible]ABC [illegible]ABC [illegible] ABC [illegible] Python [illegible] C [illegible]

[illegible]https://www.python.org/doc/essays/foreword/

[illegible]

[illegible] Python[illegible]

[illegible]

[illegible] | [illegible]

[illegible]

[illegible]——[illegible]——[illegible]

“[illegible]”[illegible]

[illegible] Python [illegible]

> “[illegible]”

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]www.github.com

[illegible] follow [illegible]: qiwsir[illegible]

[illegible] run [illegible] run[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

“[illegible]”[illegible]

[illegible] google [illegible]“[illegible]”[illegible]“[illegible]”[illegible] **google.com**[illegible]

[illegible]

[illegible]

[illegible]

Python [illegible]

[illegible] OFFICE[illegible] office [illegible]

> [illegible]
>
> [illegible] Python [illegible]
>
> “[illegible]”
>
> [illegible]“[illegible] google [illegible]”[illegible] google [illegible]
>
> [illegible] google [illegible] google[illegible]
>
> [illegible] [illegible] [illegible]

[illegible]www.Python.org/downloads/

> www.python.org [illegible] python [illegible]

□ Python □□□□□□□□□□□ Python □□□□□□□□□□ Python3.x.x□□□□□□ Python2.7.x□□□□□Python3 □□□□□□ Python2.7 □□□□□□□□□□□□□□□□□□□□□□ Python3□□□□□□□□□ Python2.7□□□ Python3□□□□□□□□□□□□□□

□□□□□□□□ Python2.7 □□□□□□□□

Linux □□□□□

□□□□□□□□□□□□□□□□□□□□□□□ Linux □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□ iOS□□□□□□□□□ UNIX □□□□□□ widows □□□□□□

□□□□□□□Python □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ Linux □□□□□□□□□□□

□□ Ubuntu□

□ ubutu14.04 □□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□ Python2.7.6 □□□□□□□□□□□□□□□□□□□□□□□□□□ Python2.7.9 □□ 64 □□□□□□□□□□□ Python3□□□□□□□□□

□□□□□ shell □□□ Python□□□□□□`>>>`□□□□□□ Python □□□□□□□□□□□□□□□□ Python □□□□□□□

□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□

  wget http://www.python.org/ftp/python/2.7.8/Python-2.7.8.tgz

- □□□□□

tar -zxvf Python-2.7.8.tgz

- □□

  cd Python-2.7.8 ./configure --prefix=/usr/local # □□□□ □□□□□□□□□□□□□□□□□□□□ ./configure □□□ make&&sudo make install

□□□□□□□□ shell□□□ Python□□□□□□□

```
 qw@qw-Latitude-E4300:~$ python
Python 2.7.6 (default, Nov 13 2013, 19:24:16)
# □□□□□□ 2.7.8 □,□□□□□□□□□□□□□□
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license"
for more information.
>>>
```

□□□□ Python2.7.6□□□□□□□□□□□□□□□□□□□□□□

windows □□□□□

□□□□□□□□ windows □□□□□□□□□□□□□□□□□□Python-2.7.8.msi□□□□□□□□“□□□”□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□ Python□

> □□□□□□□□□ windows □□□□□□□□□□□□□□□□□□ google □□□□□□"windows □□□□"□□□□□□□□□□

□□□□□□ cmd □□□□ Python□□□□□□□□□□□□□□□□□□□□□□

Mac OS X □□□□□

[illegible] Mac OS X [illegible] Mac OS X [illegible] windows [illegible] Mac OS X [illegible] unbutu [illegible]

[illegible] google[illegible] [illegible] [illegible]

[illegible]Linux [illegible] Mac OS x [illegible] Python [illegible] windows [illegible]ActivePython

[illegible] linux [illegible]

1. [illegible] root [illegible]
2. [illegible] Python: wget https://www.Python.org/ftp/Python/2.7.8/Python-2.7.8.tgz
3. [illegible]tar xfz Python-2.7.8.tgz
4. [illegible]cd Python-2.7.8
5. [illegible] ./configure
6. [illegible]make[illegible]make install
7. [illegible]
8. [illegible]

OK![illegible]

[illegible]“[illegible]”[illegible]

[illegible] Python2[illegible] Python3 [illegible] 2 [illegible] 3[illegible] Python2[illegible] Python3 [illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

[illegible](IDE)

[illegible] Python [illegible]Hello World

[illegible]Hello world

[illegible] Python[illegible]

[illegible]

```
Python 2.7.6 (default, Nov 13 2013, 19:24:16)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license"
for more information.
>>>
```

[illegible] `>>>` [illegible] `print "Hello, World"` [illegible]

```
>>> print "Hello, World"
Hello, World
```

[illegible];[illegible]

`Hello, World` [illegible]

[illegible]

> [illegible]#[illegible] [illegible] [illegible] Python [illegible]

```
# [illegible]“>>>”[illegible] Python [illegible]

>>>

# print[illegible] Python [illegible]

>>> print

#"Hello,World"[illegible] Python[illegible]

>>> print "Hello, World"

# [illegible]Python [illegible] Hello,World[illegible]

Hello, World
```

[illegible] Python [illegible]“[illegible]”[illegible]

> [illegible]Hello World

[illegible] Hello,World[illegible]“[illegible]”[illegible]

[illegible] Photoshop[illegible] word[illegible] Excel[illegible] PPT [illegible]

[illegible]“[illegible]”[illegible]

IDE [illegible]

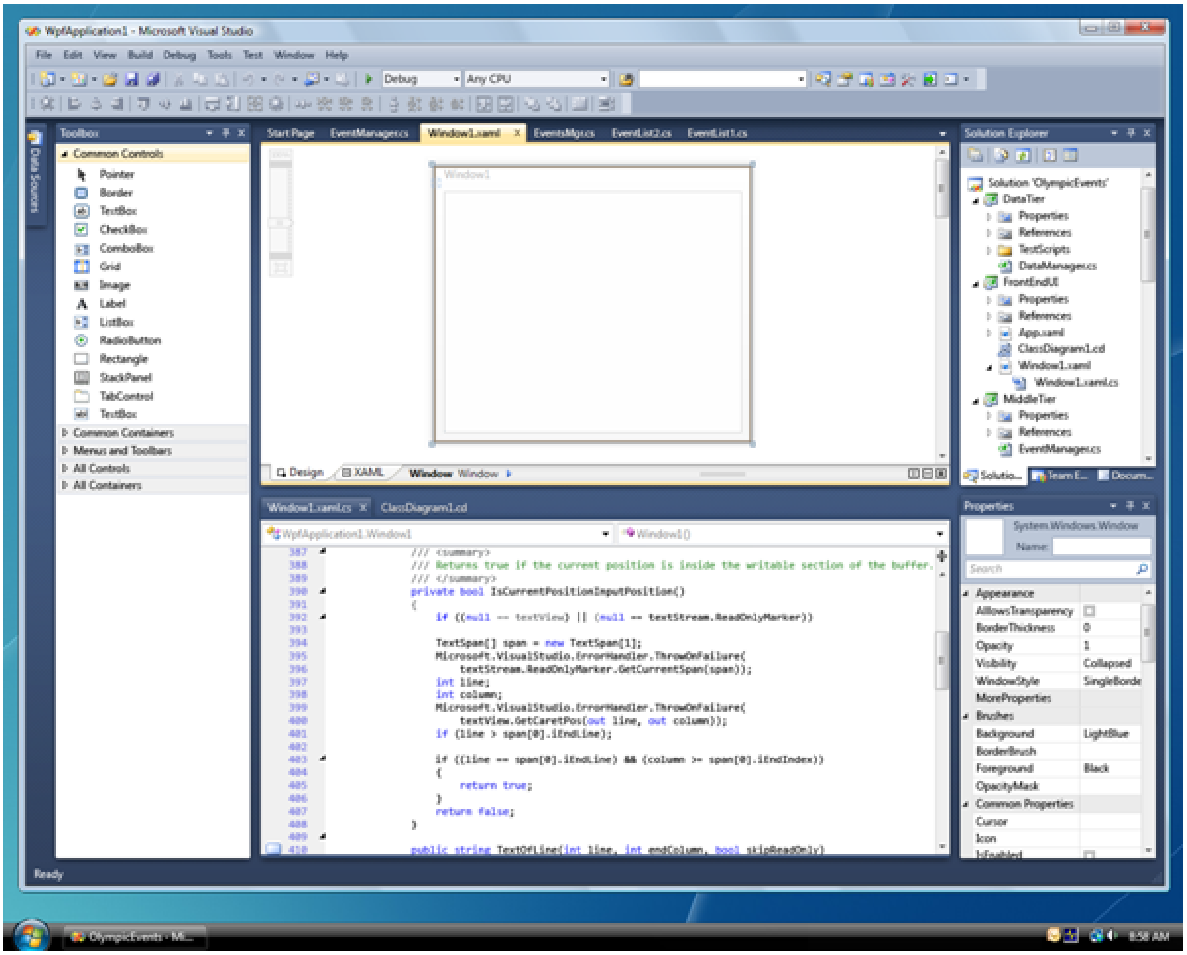
[illegible]

IDE [illegible]Integrated Development Environment[illegible] IDE[illegible] Integration Design Environment[illegible]Integration Debugging Environment[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] IDE [illegible]

> IDE [illegible] IDE [illegible] Microsoft Visual Studio[illegible] Eclipse[illegible] SharpDevelop [illegible] IDE [illegible] IDE [illegible] IDE [illegible]IDE[illegible] Eclipse[illegible]NetBeans[illegible]Microsoft Visual Studio[illegible]IDE [illegible] Visual Basic[illegible]

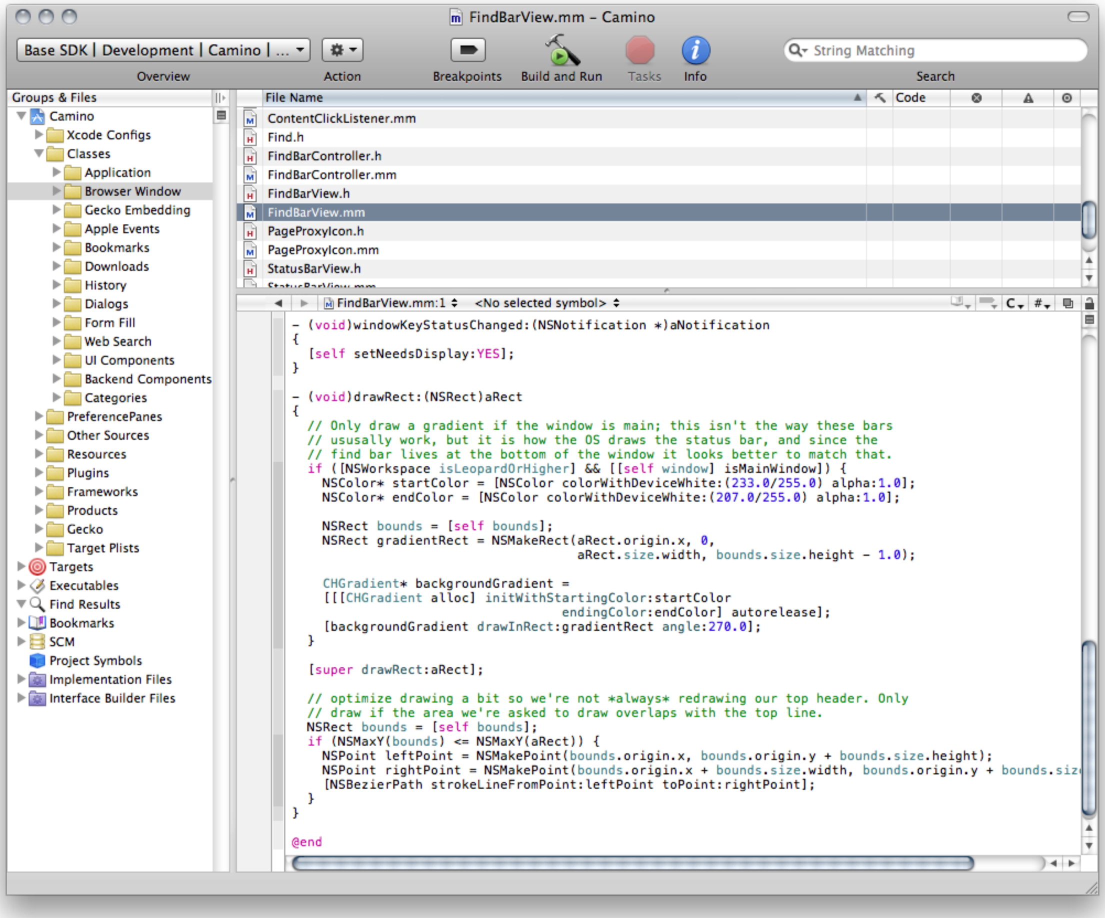
[illegible]



```
/// <summary>
/// Returns true if the current position is inside the writable section of the buffer.
/// </summary>
private bool IsCurrentPositionInputPosition()
{
    if ((null == textView) || (null == textStream.ReadOnlyMarker))

    TextSpan[] span = new TextSpan[1];
    Microsoft.VisualStudio.ErrorHandler.ThrowOnFailure(
        textStream.ReadOnlyMarker.GetCurrentSpan(span));
    int line;
    int column;
    Microsoft.VisualStudio.ErrorHandler.ThrowOnFailure(
        textView.GetCaretPos(out line, out column));
    if (line > span[0].iEndLine);

    if ((line == span[0].iEndLine) && (column >= span[0].iEndIndex))
    {
        return true;
    }
    return false;
}

public string TextOfLine(int line, int endColumn, bool skipReadOnly)
```

WpfApplication1 - Microsoft Visual Studio
File Edit View Build Debug Tools Test Window Help
Debug
Any CPU
Toolbox
Common Controls
Pointer
Border
TextBox
CheckBox
ComboBox
Grid
Image
Label
ListBox
RadioButton
Rectangle
StackPanel
TabControl
TextBox
Common Containers
Menus and Toolbars
All Controls
All Containers
Data Sources
Start Page
EventManager.cs
Window1.xaml
EventsMgr.cs
EventList2.cs
EventList1.cs
Window1
Design
XAML
Window Window
Window1.xaml.cs
ClassDiagram1.cd
WpfApplication1.Window1
Window1()
Solution Explorer
Solution 'OlympicEvents'
DataTier
Properties
References
TestScripts
DataManager.cs
FrontEndUI
Properties
References
App.xaml
ClassDiagram1.cd
Window1.xaml
Window1.xaml.cs
MiddleTier
Properties
References
EventManager.cs
Solutio...
Team E...
Docum...
Properties
System.Windows.Window
Name:
Search
Appearance
AllowsTransparency
BorderThickness 0
Opacity 1
Visibility Collapsed
WindowStyle SingleBorde
MoreProperties
Brushes
Background LightBlue
BorderBrush
Foreground Black
OpacityMask
Common Properties
Cursor
Icon
IsEnabled
Ready
OlympicEvents - Mi...
8:58 AM

□□□□□□□□□□□□□□□□□□□□ Microsoft Visual Studio □ IDE□□ C# □
□□□□□□□□□□□□


FindBarView.mm – Camino
Base SDK | Development | Camino | ...
Overview
Action
Breakpoints
Build and Run
Tasks
Info
String Matching
Search
Groups & Files
Camino
Xcode Configs
Classes
Application
Browser Window
Gecko Embedding
Apple Events
Bookmarks
Downloads
History
Dialogs
Form Fill
Web Search
UI Components
Backend Components
Categories
PreferencePanes
Other Sources
Resources
Plugins
Frameworks
Products
Gecko
Target Plists
Targets
Executables
Find Results
Bookmarks
SCM
Project Symbols
Implementation Files
Interface Builder Files
File Name
Code
ContentClickListener.mm
Find.h
FindBarController.h
FindBarController.mm
FindBarView.h
FindBarView.mm
PageProxyIcon.h
PageProxyIcon.mm
StatusBarView.h
FindBarView.mm:1
<No selected symbol>

```
- (void)windowKeyStatusChanged:(NSNotification *)aNotification
{
  [self setNeedsDisplay:YES];
}

- (void)drawRect:(NSRect)aRect
{
  // Only draw a gradient if the window is main; this isn't the way these bars
  // ususally work, but it is how the OS draws the status bar, and since the
  // find bar lives at the bottom of the window it looks better to match that.
  if ([NSWorkspace isLeopardOrHigher] && [[self window] isMainWindow]) {
    NSColor* startColor = [NSColor colorWithDeviceWhite:(233.0/255.0) alpha:1.0];
    NSColor* endColor = [NSColor colorWithDeviceWhite:(207.0/255.0) alpha:1.0];

    NSRect bounds = [self bounds];
    NSRect gradientRect = NSMakeRect(aRect.origin.x, 0,
                                     aRect.size.width, bounds.size.height - 1.0);

    CHGradient* backgroundGradient =
    [[[CHGradient alloc] initWithStartingColor:startColor
                                   endingColor:endColor] autorelease];
    [backgroundGradient drawInRect:gradientRect angle:270.0];
  }

  [super drawRect:aRect];

  // optimize drawing a bit so we're not *always* redrawing our top header. Only
  // draw if the area we're asked to draw overlaps with the top line.
  NSRect bounds = [self bounds];
  if (NSMaxY(bounds) <= NSMaxY(aRect)) {
    NSPoint leftPoint = NSMakePoint(bounds.origin.x, bounds.origin.y + bounds.size.height);
    NSPoint rightPoint = NSMakePoint(bounds.origin.x + bounds.size.width, bounds.origin.y + bounds.siz
    [NSBezierPath strokeLineFromPoint:leftPoint toPoint:rightPoint];
  }
}

@end
```

□□□□□□□□□□□□□□ XCode □ IDE□

□□□□□□ IDE □□□□□□□□□□□□□□□□

- □□□□□Integrated development environment
- □□□□□□□□□□□

Python □ IDE

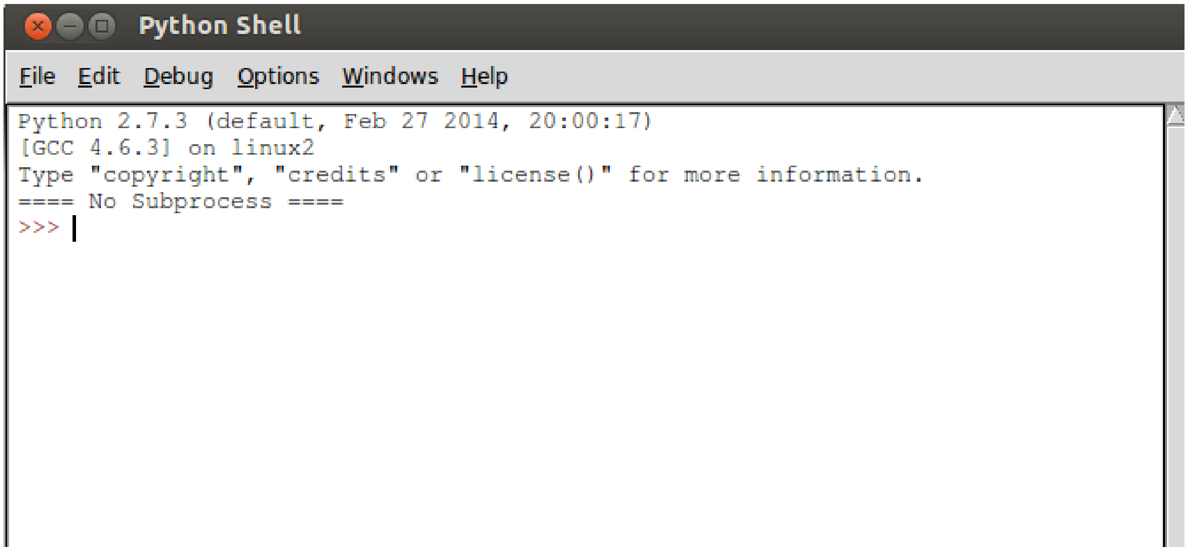
google □□□Python IDE□□□□□□□□□ Python □□□ IDE □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ IDE □□□□□□□□□□□□□□□□□ IDE □□□□

> □□□□□□□□□□□□□□□□□□□□□□stackoverflow.com
>
> □□
>
> □□

□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ Python □□□ IDLE□□□□□□□□□

Windows □□□□□□“□□”□□->“□□□□□”->“Python 2.x”->“IDLE □Python GUI□”□□□ IDLE□□□□□□□□□□□□□□□



Python Shell
File
Edit
Debug
Options
Windows
Help

```
Python 2.7.3 (default, Feb 27 2014, 20:00:17)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
>>> 
```

□□□

□□□□□□□□□□□□□□□□ idle □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ IDE□

□□□□□□□ IDE□□□□□□□□ IDE□□□□□□□□□□□□□□□

- PythonWin: □ Python Win32 Extensions(□□□□□□ Python for win32 □□□)□□□□□□□□□ ActivePython □

windows [illegible] win32 [illegible]

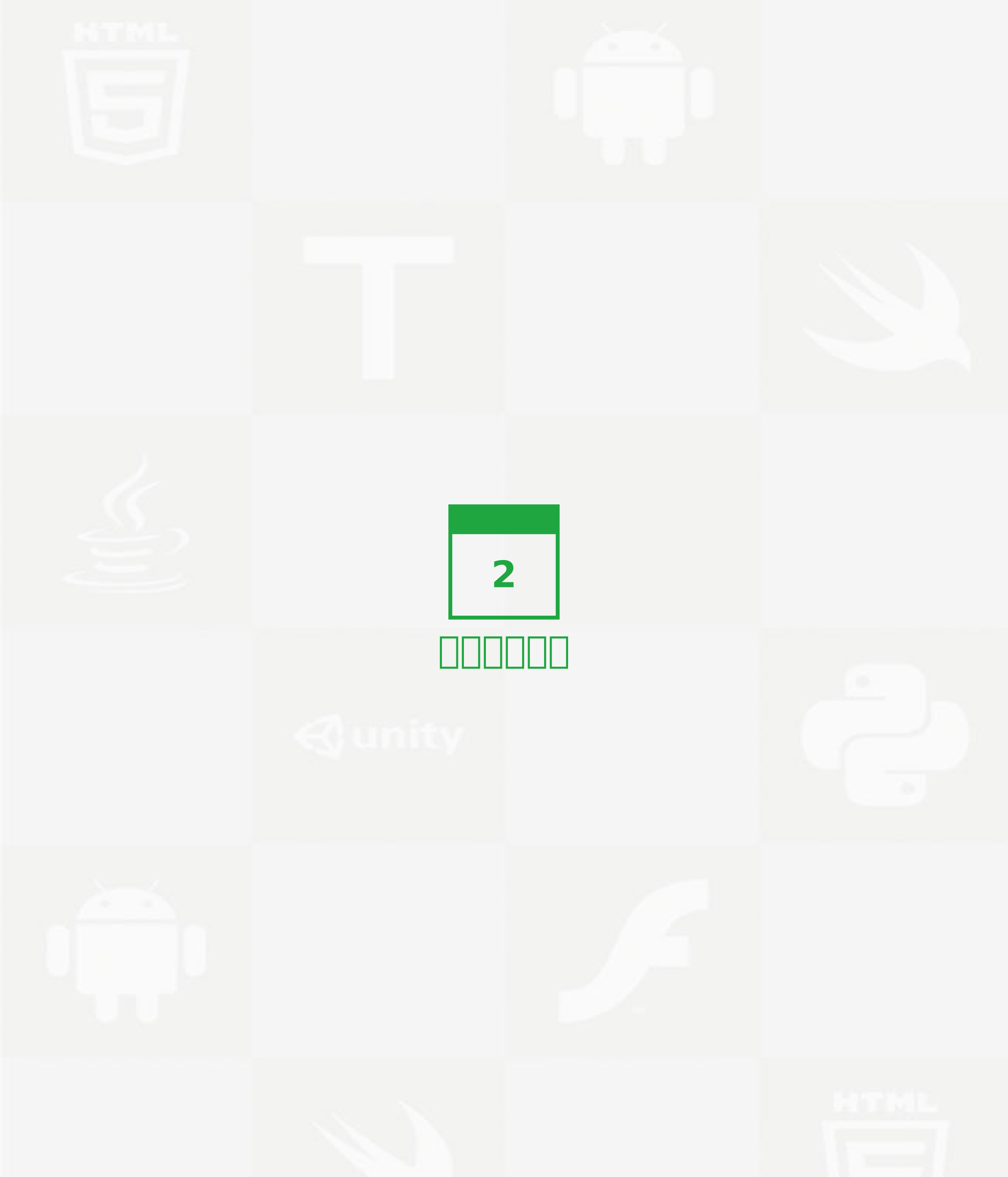
- MacPython IDE: MacPythonIDE [illegible] Python [illegible] Mac OS [illegible] IDE[illegible] PythonWin [illegible] Mac [illegible] Guido [illegible] Just van Rossum [illegible]([illegible])
- Emacs [illegible] Vim: Emacs [illegible] Vim [illegible]([illegible])[illegible] IDE [illegible]([illegible]?)[illegible]
- Eclipse + PyDev: Eclipse [illegible] IDE[illegible] Java [illegible] Emacs [illegible] Vim [illegible]

[illegible] IDE[illegible]

[illegible]IDE [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]



2

□□□□□□

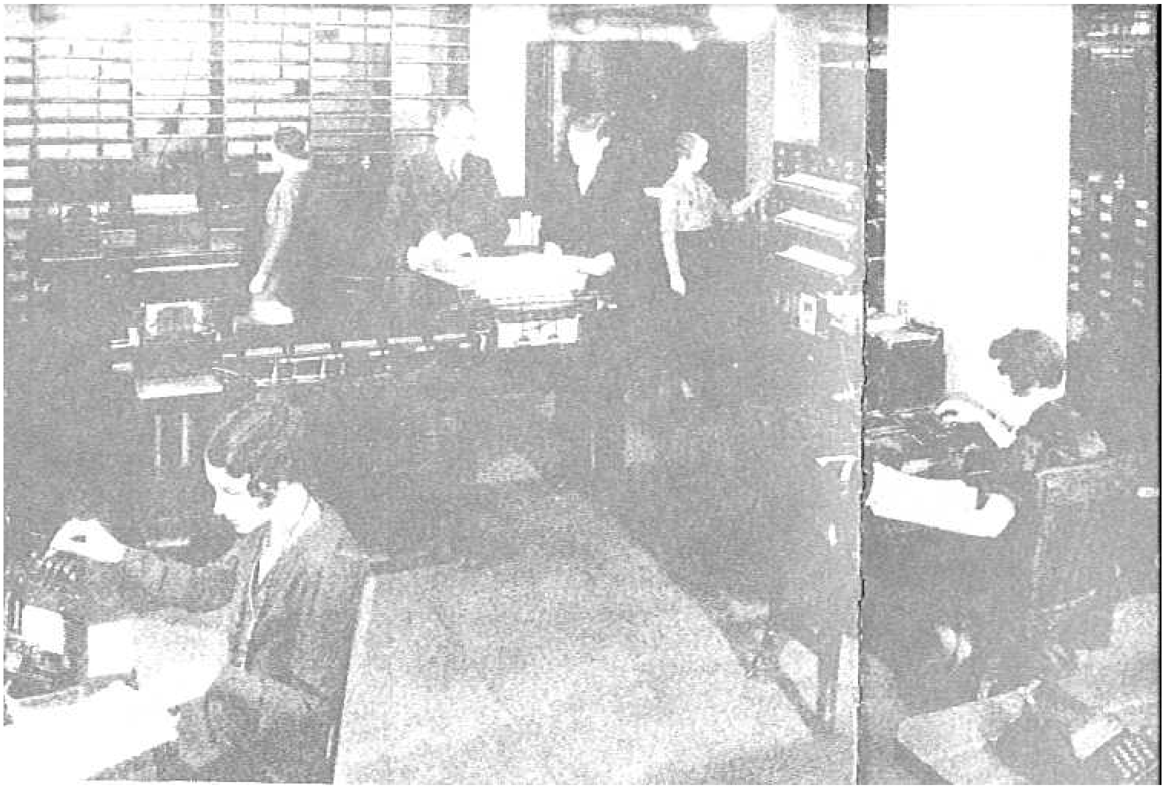
[illegible]

[illegible] computer[illegible]

[illegible]

> [illegible]computer[illegible] [illegible]computer [illegible]“compute-er”[illegible]“[illegible]”[illegible] [illegible] [illegible] computer [illegible]“[illegible]”[illegible] computer[illegible]

[illegible]



[illegible] computer[illegible] she[illegible]“[illegible]”[illegible] computer [illegible]“[illegible]”[illegible]

[illegible]

[illegible] Python [illegible]

[illegible]

```
>>> 3
3
>>> 3333333333333333333333333333333333333333
3333333333333333333333333333333333333333L
>>> 3.222222
3.222222
```

[illegible] 3[illegible] 3[illegible]

[illegible] 3 [illegible] Python [illegible]Python [illegible]L[illegible] Python [illegible]

[illegible]“[illegible]”[illegible] google.

[illegible]

[illegible]

[illegible]

[illegible] Python [illegible] 3[illegible]

```
>>> id(3)
140574872
>>> id(3.222222)
140612356
>>> id(3.0)
140612356
>>>
```

[illegible] id()[illegible]

> [illegible] built-in Function[illegible] Python [illegible]

[illegible]3 [illegible] 3.0 [illegible]

[illegible] id()[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]type()[illegible]

```
>>> type(3)
<type 'int'>
>>> type(3.0)
<type 'float'>
>>> type(3.222222)
<type 'float'>
```

□□□□□□□□□□□□□□□□□□ 3 □□□□□□Interger□□□□□□□□□□□□□□□□Floating point real number□□□ id()□□□□□□type()□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□

□□

□□□□ 3□4□5 □□□□□□□□□□□□□□□□□□□“□□”□“□”□□ Python □□□□□□□□□□□□□□□□□□□□□

```
>>> x = 5
>>> x
5
>>> x = 6
>>> x
6
```

□□□□□□□`x = 5`□□□□□(x)□□(5)□□□□□□□□□□□□□□□□ x □ 6 □□□□□□□□□□□□□□□x□“□”5□□□“□”6□

□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□5□6 □□□□□Python □□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□ int □□□

□□□□ Python □□□□ 5□6□computer □□ 5□□□□□□ 6□□□□□□□□□□□□□□□□□□ int.

□□ x □□□□□□□□□□□□`x = 5`□□□□□ x □□□□□□□ 5 □□□□□□□ x□□□□□□□ 5□□□□□□□□□□`>>> x`□□□□□□□ 5□□□□□□□□□ x □□ 5□□□□ x □□□□□□ 5□□□□□□□□□□`x = 6`□□□□□□□□□□□□ 6 □□□

□□□□□□□ x □□□ str□□□□□□□□□□□□□

□□ Python □□□□□□□□□□□□□□□

[illegible]

[illegible]

```
>>> 2+5
7
>>> 5-2
3
>>> 10/2
5
>>> 5*2
10
>>> 10/5+1
3
>>> 2*3-4
2
```

[illegible](+)[illegible](-)[illegible](*)[illegible](/)

[illegible]

[illegible]

[illegible]

[illegible]

- 4 + 2
- 4.0 + 2
- 4.0 + 2.0

[illegible]

[illegible]

```
>>> 4+2
6
>>> 4.0+2
6.0
>>> 4.0+2.0
6.0
```

[illegible] 6[illegible] 6.0[illegible]

> □□ 1□□□ 4□-2□129486655□-988654□0 □□□□□□□□□□□□
> □□ 2□□□ 4.0□-2.0□2344.123□3.1415926 □□□□□□□□□□□□□

□□□□□□□□□□□□□□□google □□□ □□□□□□□□□□□□□□□□□□□□□□ google□

□□□ □□□□□□□□□□□ google□□□ □□□□□□□□□□□□□□□□□□□□“□□□□□□”□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□ □□□[□□□□□href="https://github.com/qiwsir/algorithm/blob/master/big_int.md")

ok!□□□□□ IDE □□□□□□□□□□□

```
>>>
123456789870987654321122343445567678890098876*1233455667789990099876543332387665443345566
152278477193527562870044352587576277277562328362032444339019158937017801601677976183816L
```

□□□□□□□Python □□□□□□□□□□□ Python □□□□□□□□

□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ google□□□□□□□□□□□□□

- □□□□ int □□□□□□□□integer
- □□□□□ float □□□□□□□□float

□□□□□□□□type(object)□□□□□□□□□□□□

```
>>> type(4)
<type 'int'>    #4 □ int□□□
>>> type(5.0)
<type 'float'>  #5.0 □ float□□□□
type(988776544222112233445566778899887766554433221133344455566677788998776543222344556678)
<type 'long'>   # □□□□□□□□□□□
```

□□□

[illegible]qiwsir@126.com,[illegible]

[illegible]

[illegible] Python[illegible]

[illegible]

[illegible] Python [illegible]>>>[illegible]

```
>>> 2 / 5
0
>>> 2.0 / 5
0.4
>>> 2 / 5.0
0.4
>>> 2.0 / 5.0
0.4
```

[illegible] Python2.x [illegible] 0.4[illegible] 0[illegible]why?

[illegible] Python[illegible] Python2.x [illegible]Python3 [illegible] 2/5 [illegible]2 [illegible] 5[illegible] 0[illegible] 2[illegible]2/5[illegible]

[illegible]

```
>>> 5 / 2
2
>>> 7 / 2
3
>>> 8 / 2
4
```

[illegible],[illegible]“[illegible]”[illegible]5/2[illegible] 2[illegible] 1[illegible]`5 / 2 = 2`[illegible] 2.5 [illegible]

[illegible]

□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□□”□□□□□
□□□“□□□□□□□□”□□□□□□

> □□□x □□ y□□□ x □□□□□□□□□□□□□□□y □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

```
>>> 9.0 / 2
4.5
>>> 9 / 2.0
4.5
>>> 9.0 / 2.0
4.5

>>> 8.0 / 2
4.0
>>> 8 / 2.0
4.0
>>> 8.0 / 2.0
4.0
```

□□□

```
>>> 10.0 / 3
3.3333333333333335
```

□□□□□□□□□□□□□□□□□□□□□□□□ 3.33333...□□□□ 3 □□□□□□□□□□□□□□□□□□□□□□□□□ 3□□□□□□□□Python □□□□□□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□

```
>>> 0.1 + 0.2
0.30000000000000004
>>> 0.1 + 0.1 - 0.2
0.0
>>> 0.1 + 0.1 + 0.1 - 0.3
5.551115123125783e-17
>>> 0.1 + 0.1 + 0.1 - 0.2
0.10000000000000003
```

□□□□□□□□□□□ computer □□□□□□□□□□□□□□□□□□□□□□□□□ computer □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□computer □□

□□□□□□ 0.1□□□□□□□□□□
0.0001100110011001100110011001100110011001100110011...

□□□□□□□□□□□□□□□□□□□□□□□□□ 0.1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible] Python [illegible] Python [illegible] bug[illegible]

[illegible] Python [illegible] 2**53 [illegible]

[illegible]“[illegible]”[illegible]

[illegible] decimal [illegible] fractions [illegible] 1/3 [illegible] SciPy [illegible] Numerical Python [illegible]

[illegible]0.999...[illegible]

> [illegible]

Python [illegible]

[illegible]--[illegible]

Python [illegible]...[illegible]

[illegible]

Python [illegible] Python [illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

- [illegible] 1[illegible]import module-name[illegible]import [illegible]import os
- [illegible] 2[illegible]from module1 import module11[illegible]module1 [illegible] module11[illegible] module11[illegible]

[illegible]

```
>>> from __future__ import division
>>> 5 / 2
2.5
>>> 9 / 2
4.5
>>> 9.0 / 2
4.5
>>> 9 / 2.0
4.5
```

[illegible]

[illegible]

[illegible]

[illegible] 5/2 [illegible] 2[illegible] 1

[illegible] Python [illegible] `%` [illegible].

[illegible]

```
>>> 5 % 2
1
>>> 6%4
2
>>> 5.0%2
1.0
```

[illegible]%[illegible]

[illegible] Python [illegible] `divmod()`

```
>>> divmod(5,2)  # [illegible] 5 [illegible] 2[illegible]
(2, 1)
>>> divmod(9,2)
(4, 1)
>>> divmod(5.0,2)
(2.0, 1.0)
```

[illegible]

[illegible] `round()`

[illegible]

```
>>> round(1.234567,2)
1.23
>>> round(1.234567,3)
1.235
>>> round(10.0/3,4)
3.3333
```

[illegible]

```
>>> round(1.2345,3)
1.234               # □□□□1.235
>>> round(2.235,2)
2.23                # □□□□2.24
```

□□□□□□□ Python □□□ bug□□□□□□

□□□□□□□□□□□ bug□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> **Note:** The behavior of round() for floats can be surprising: for example, round(2.675, 2) gives 2.67 instead of the expected 2.68. This is not a bug: it’s a result of the fact that most decimal fractions can’t be represented exactly as a float. See Floating Point Arithmetic: Issues and Limitations for more information.

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ github □□□□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

[illegible]

[illegible]——[illegible]——[illegible] Python [illegible]Math

> [illegible](module)[illegible] Python [illegible] Python [illegible]Python [illegible]“[illegible]”
> [illegible] Pythoner [illegible]
> [illegible] Python [illegible]“[illegible]”[illegible]“[illegible]”[illegible]
>
> [illegible] pip [illegible] google[illegible]

[illegible] **math** [illegible]

math [illegible]

```
>>> import math
```

[illegible] import [illegible] math [illegible]

```
>>> math.pi
3.141592653589793
```

[illegible]

```
>>> dir(math)
['  doc  ', '  name  ', '  package  ', 'acos', 'acosh', 'asin',
'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',
'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',
'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'hypot',
'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p',
'modf', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan',
'tanh', 'trunc']
```

`dir(module)` [illegible] math [illegible] sin(a),cos(a),sqrt(a)......

[illegible] math [illegible] pow [illegible]

Python [illegible]

```
>>> help(math.pow)
```

[illegible]

```
Help on built-in function pow in module math:

pow(...)
    pow(x, y)

    Return x**y (x to the power of y).
```

[illegible] math [illegible] pow [illegible]

1. [illegible] math [illegible] pow [illegible] built-in[illegible] Python [illegible])
2. [illegible]
3. [illegible] `x**y` [illegible] `x**y` [illegible]
4. [illegible] q [illegible] Python [illegible]

[illegible] pow [illegible] `x**y` [illegible] x [illegible] y [illegible]

```
>>> 4**2
16
>>> math.pow(4,2)
16.0
>>> 4*2
8
```

[illegible]`4**2` [illegible] `4*2` [illegible]

[illegible] math [illegible]

> [illegible]“[illegible]”[illegible]

[illegible] math [illegible]

```
>>> math.sqrt(9)
3.0
>>> math.floor(3.14)
3.0
>>> math.floor(3.92)
3.0
>>> math.fabs(-2)    # [illegible] abs(-2)
2.0
>>> abs(-2)
```

```
2
>>> math.fmod(5,3)    # □□□ 5%3
2.0
>>> 5%3
2
```

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ google□

□□□□

```
 >>> abs(10)
10
>>> abs(-10)
10
>>> abs(-1.2)
1.2
```

□□□□

```
 >>> round(1.234)
1.0
>>> round(1.234,2)
1.23

>>> # □□□□□□□□□□□□□□□□□□□□□□□□□□
>>> help(round)

Help on built-in function round in module __builtin__:

round(...)
    round(number[, ndigits]) -> floating point number

    Round a number to a given precision in decimal digits
(default 0 digits).
    This always returns a floating point number.  Precision may
be negative.
```

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□”□□□□□□□□□□□□□□□□□□

[illegible]“[illegible]”[illegible]

[illegible] Python [illegible]

[illegible]	[illegible]
lambda	Lambda [illegible]
or	[illegible]“[illegible]”
and	[illegible]“[illegible]”
not x	[illegible]“[illegible]”
in[illegible]not in	[illegible]
is[illegible]is not	[illegible]
<[illegible]<=[illegible]>[illegible]>=[illegible]!=[illegible]==	[illegible]
|	[illegible]
^	[illegible]
&	[illegible]
<<[illegible]>>	[illegible]
+[illegible]-	[illegible]
*[illegible]/[illegible]%	[illegible]
+x[illegible]-x	[illegible]
~x	[illegible]
**	[illegible]
x.attribute	[illegible]
x[index]	[illegible]
x[index:index]	[illegible]
f(arguments...)	[illegible]
(experession,...)	[illegible]

[illegible]	[illegible]
[expression,...]	[illegible]
{key:datum,...}	[illegible]
'expression,...'	[illegible]

[illegible] Python [illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible]

[illegible] Python [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

- [illegible]computer program and source code[illegible]

> A computer program, or just a program, is a sequence of instructions, written to perform a specified task with a computer.[1] A computer requires programs to function, typically executing the program's instructions in a central processor.[2] The program has an executable form that the computer can use directly to execute the instructions. The same program in its human-readable source code form, from which executable programs are derived (e.g., compiled), enables a programmer to study and develop its algorithms. A collection of computer programs and related data is referred to as the software.
>
> Computer source code is typically written by computer programmers.[3] Source code is written in a programming language that usually follows one of two main paradigms: imperative or declarative programming. Source code may be converted into an executable file (sometimes called an executable program or a binary) by a compiler and later executed by a central processing unit. Alternatively, computer programs may be executed with the aid of an interpreter, or may be embedded directly into hardware.
>
> Computer programs may be ranked along functional lines: system software and application software. Two or more computer programs may run simultaneously on one computer from the perspective of the user, this process being known as multitasking.

- [illegible]

> □□□□□□Computer Program□□
> □□□
> □□□□□□□□□□□□□□□□□
>
> □□□
> □□script□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Python □□□“□□□□□□□□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□“□□”□□

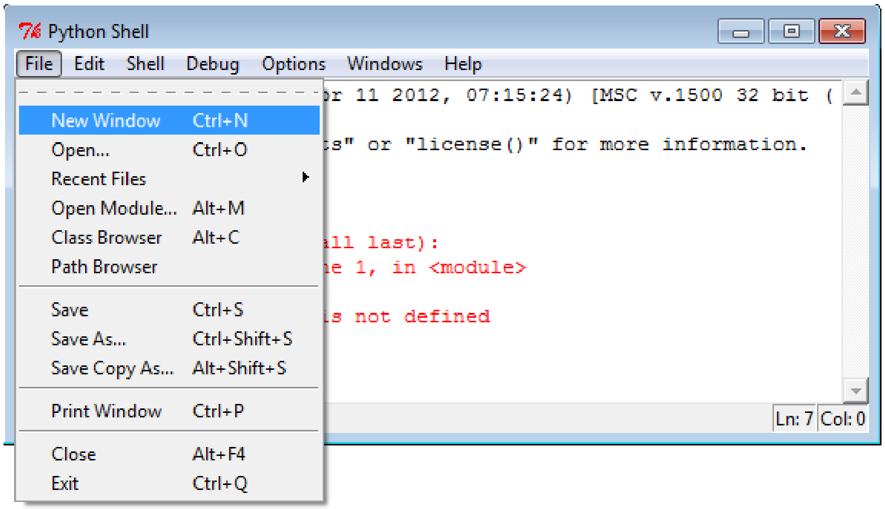
□□□□□□□□□□□

□ IDLE □□□□□

□□□ Python □□Python □□ vim□□□□□ eclipse□□□ notepad++□□□□□□□□□□□□□□□□□□□□□□

□ Python □□□ IDE □□□□□□□□

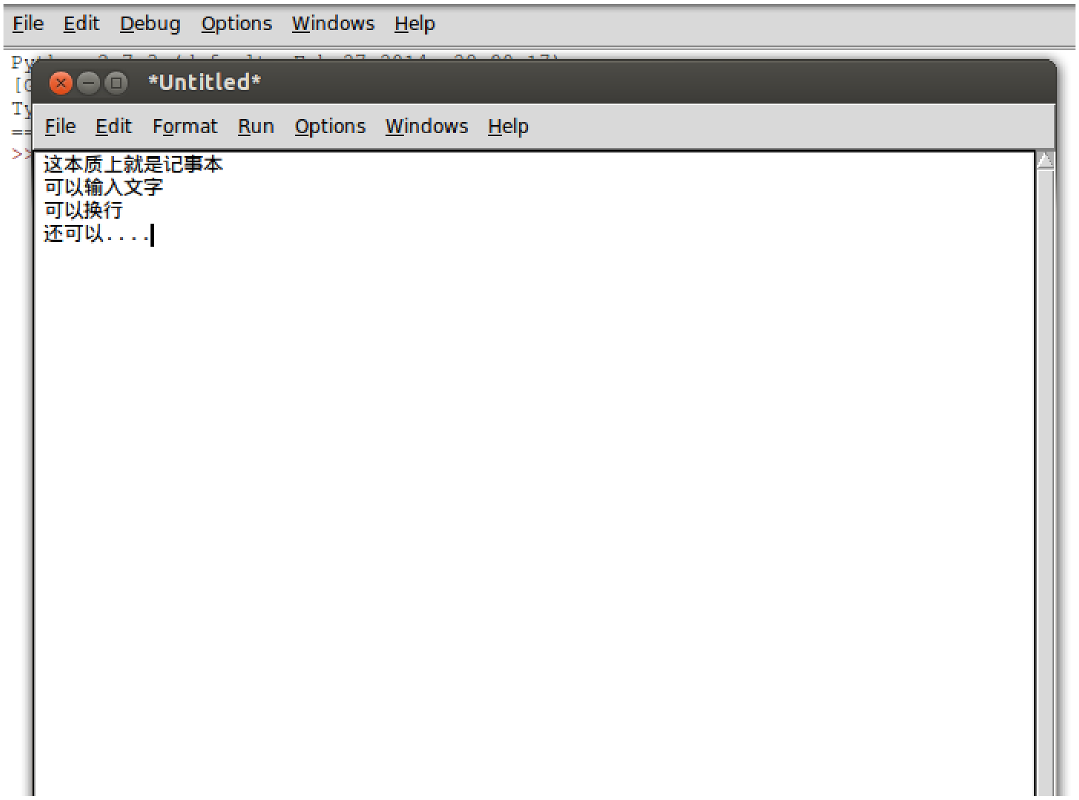
□□□File->New window



Python Shell
File
Edit
Shell
Debug
Options
Windows
Help
New Window Ctrl+N
Open... Ctrl+O
Recent Files
Open Module... Alt+M
Class Browser Alt+C
Path Browser
Save Ctrl+S
Save As... Ctrl+Shift+S
Save Copy As... Alt+Shift+S
Print Window Ctrl+P
Close Alt+F4
Exit Ctrl+Q
Ln: 7
Col: 0

```
or 11 2012, 07:15:24) [MSC v.1500 32 bit (

s" or "license()" for more information.



all last):
e 1, in <module>

s not defined
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□>>>□□□


File Edit Debug Options Windows Help
Untitled
File Edit Format Run Options Windows Help

```
这本质上就是记事本
可以输入文字
可以换行
还可以....
```

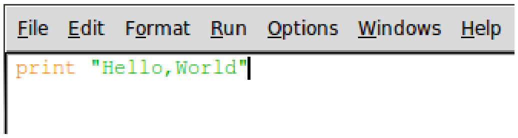
□□□□□□Hello,World

Hello,World.□□ □□□□□□□□□□□□□□□□□□□□ Hello□

□□□□□□□□□□□□□□

```
print "Hello,World"
```

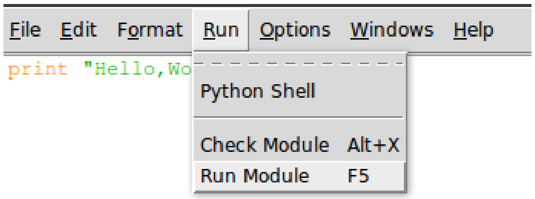
□□□□□□


File Edit Format Run Options Windows Help

```
print "Hello,World"
```

□□

□□□□□□□□□□□ RUN□□□□□□□□□□□□□□□□□ Run Module□


File
Edit
Format
Run
Options
Windows
Help
Python Shell
Check Module Alt+X
Run Module F5

```
print "Hello,Wo
```

□□.py □□□□□

□□□□□□□□□□□□□□□□□□□ >>> □□□□□□□□ Hello,World □□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□ IDLE □□□□□□□□□□□□□□□□□□□□□□□

□□□□ IDLE□□□ shell(□□□□□□□□ Mac OS □□□□□□□ linux □□□□□□□□□□□□□ ubuntu)□□□□ cmd(windows □□□□□□□□□□ windows □□□□□ windows □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ linux □□□□□□□□□□□□)□□□□□□ □□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□ 105.py□□□□□□□□□□

```
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ ls
105.py
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$
```

□□□□ shell □□□□□Python 105.py

□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□ 105.py

□□□□□□□□□□□□□□□□□□□□□□□

```
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ ls
105.py
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ python 105.py
Hello,World
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ 
```

□□□□□□□□□□□ 7:1 □□

□□□□

□□□19+2*4-8/2

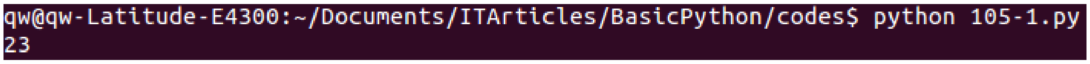
□□□□

```
#!/usr/bin/env python
#coding:utf-8

"""
□□□□
19+2*4-8/2
"""

a = 19+2*4-8/2
print a
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□:105-1.py)□

□ shell □□ cmd □□□□□Python (□□□.py)

□□□□□□□□



```
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ python 105-1.py
23
```

□□□□□□□□□

□□□□□□□□□□□□□□□□

```
#!/usr/bin/env python
```

□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□

```
#coding:utf-8
```

□□□□□□ Python□□□□□□□□□□□□ utf-8□□□□□□□□□□ utf-8□□□□□□□□□□□□□□□□□□□□□ □□

```
"""
□□□□
19+2*4-8/2
"""
```

□□□□□□□□□□□□□□□□□ Python □□

□□□□□□□□□□□□□□□□□□□ # □□□□□□□□□□□□□□□□'''□□□□□□□□

```
"""
□□□□
19+2*4-8/2
"""
```

[illegible] `#` [illegible]

```
# [illegible]19+2*4-8/2
```

[illegible] `#` [illegible]

[illegible]

```
a = 19+2*4-8/2
```

[illegible]`19+2*4-8/2` [illegible]“[illegible]”[illegible]`=` [illegible]“[illegible]”[illegible]`a` [illegible]

> [illegible]“[illegible]”[illegible]“[illegible]”[illegible]

```
print a
```

[illegible] print [illegible] a [illegible] a [illegible] a [illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible](1)

[illegible]“[illegible]”[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible] 3 [illegible] 5 [illegible] 8[illegible] 35[illegible]

[illegible]

- [illegible]△ + □ = ○
- [illegible]△ + □ = △ □

[illegible]

[illegible]

[illegible]

[illegible] google [illegible]

> [illegible]String[illegible] s=a[1]a[2]...a[n][illegible]

[illegible]

[illegible]"Hello,World"[illegible]

[illegible] Python [illegible] str [illegible]`''`[illegible]`""`[illegible]

> [illegible]

```
>>> "I love Python."
'I love Python.'
>>> 'I LOVE PYTHON.'
'I LOVE PYTHON.'
```

[illegible]

```
>>> 250
250
>>> type(250)
<type 'int'>

>>> "250"
'250'
>>> type("250")
<type 'str'>
```

[illegible] 250[illegible] `type()` [illegible] int [illegible] str [illegible] int[illegible]or float[illegible],[illegible] `type()` [illegible]

[illegible]

```
>>> print "good good study, day day up"
good good study, day day up
>>> print "----good---study---day----up"
----good---study---day----up
```

[illegible] print [illegible]

[illegible]

```
What's your name?
```

[illegible]

```
>>> 'What's your name?'
File "<stdin>", line 1
 'What's your name?'
      ^
SyntaxError: invalid syntax
```

[illegible] `SyntaxError` [illegible] `SyntaxError` [illegible]“invalid syntax”[illegible] ^ [illegible]

> [illegible] python [illegible] google [illegible]

[illegible]computer [illegible]

[illegible]

```
>>> "What's your name?"
"What's your name?"
```

□□

□□□□□□□□

□□ Python □□ `\` □□□□□□□□□□□□□□□□□□□□□□□

```
>>> 'What\'s your name?'
"What's your name?"
```

□□□□□□ `\` □□□□

□□

□□□□

□□□□□□□□□□□□□□□□□□:

```
>>> a = 5
>>> a
5
```

□□□□□□ a □□□□□□□□□□ 5 □□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> b = "hello,world"
>>> b
'hello,world'
>>> print b
hello,world
```

□□□□□□□□ type □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> type(a)
<type 'int'>
>>> type(b)
<type 'str'>
```

□□□□□□□□□□□a□□□□□□□ b □□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□Python □□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□3+5□□□□□ 8□□□□□□□□□□□□□□□□□□

```
>>> "Py" + "thon"
'Python'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□ □□□□□□□□□□

```
>>> "Py" - "thon"     # □□□□□□□□□□□□
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

□ `+` □□□□□□□□□□□□□□□□□□□□□□

```
>>> a = 1989
>>> b = "free"
>>> print b+a
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: cannot concatenate 'str' and 'int' objects
```

> □□□□□□□□`print`□□□□□□□□□□□□□□□□□□□□□□ Python2 □□□□□□ Python3 □□□□□□□□□□□ `print(b+a)`□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□`cannot concatenate 'str' and 'int' objects`□□ `a` □□□□□□ `int` □□□□□□□ `str` □□□□□□□□

□□□ `+` □□□ □□□□□

□□□□□□□□□□□□□

```
>>> print b + `a`
free1989
```

□□`\` □□□□□□□□□□□□□□□□1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ Python3 □□□

□□□“□□□□□□□□□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□

```
>>> print b + str(a)
free1989
```

□ `str(a)` □□□□□□□□□□□□□□ str □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□ int □□□□□□□□

```
>>> a = "250"
>>> type(a)
<type 'str'>
>>> b = int(a)
>>> b
250
>>> type(b)
<type 'int'>
```

> □□□□□□□□□ int □ str □□□□□□□□□□□□□□□□ help(int)□help(str)□□□□□□□□□□

□□□□□□

```
>>> print b + repr(a)   #repr(a)□□□□□□
free1989
```

□□ repr()□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ python □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□repr()□ `\` □□□□□□□□□□□□□□□□□□□□□□ repr()□ str□□□□□□□□□□repr □□□□str □□ int □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ google □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> 1. When should i use str() and when should i use repr() ?
>
> Almost always use str when creating output for end users.
>
> repr is mainly useful for debugging and exploring. For example, if you suspect a string has non printing characters in it, or a float has a small rounding error, repr will show you; str may not.
>
> repr can also be useful for for generating literals to paste into your source code. It can also be used for persistence (with ast.literal_eval or eval), but this is rarely a good idea--if you want editable persisted values, something like JSON or YAML is much better, and if you don't plan to edit them, use pickle.
>
> 2.In which cases i can use either of them ?
> Well, you can use them almost anywhere. You shouldn't generally use them except as described above.
>
> 3.What can str() do which repr() can't ?
> Give you output fit for end-user consumption--not always (e.g., str(['spam', 'eggs']) isn't likely to be anything you want to put in a GUI), but more often than repr.
>
> 4.What can repr() do which str() can't
> Give you output that's useful for debugging--again, not always (the default

> for instances of user-created classes is rarely helpful), but whenever possible.
>
> And sometimes give you output that's a valid Python literal or other expression--but you rarely want to rely on that except for interactive exploration.

□□□□□□□□http://stackoverflow.com/questions/19331404/str-vs-repr-functions-in-python-2-7-5

Python □□□□

□□

□□□□	□□
\	(□□□□) □□□
\	□□□□□
\'	□□□
\"	□□□
\a	□□
\b	□□(Backspace)
\e	□□
\000	□
\n	□□
\v	□□□□□
\t	□□□□□
\r	□□
\f	□□
\oyy	□□□□□yy □□□□□□□□□\o12 □□□□
\xyy	□□□□□□yy □□□□□□□□□\x0a □□□□

[illegible]	[illegible]
\other	[illegible]

[illegible] print [illegible]

```
>>> print "hello.I am qiwsir.\                    # [illegible]
... My website is 'http://qiwsir.github.io'."
hello.I am qiwsir.My website is 'http://qiwsir.github.io'.

>>> print "you can connect me by qq\\weibo\\gmail"  #\\ [illegible]
\
you can connect me by qq\weibo\gmail
```

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□□(2)

raw_input □ print

□□□□□□□□□□□□□□□□□□ computer □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> □□□

□□ Python □□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□

□□□□□□□□□□□□□□□□□□raw_input □ print

> □□□□□ Python □□□□□□built-in function□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□

Built-in Functions

abs()	**divmod()**	**input()**	**open()**	**staticmethod()**
all()	enumerate()	int()	ord()	str()
any()	eval()	isinstance()	pow()	sum()
basestring()	execfile()	issubclass()	print()	super()
bin()	file()	iter()	property()	tuple()
bool()	filter()	len()	range()	type()
bytearray()	float()	list()	raw_input()	unichr()
callable()	format()	locals()	reduce()	unicode()
chr()	frozenset()	long()	reload()	vars()
classmethod()	getattr()	map()	repr()	xrange()
cmp()	globals()	max()	reversed()	zip()
compile()	hasattr()	memoryview()	round()	**import**()

abs()	**divmod()**	**input()**	**open()**	**staticmethod()**
complex()	hash()	min()	set()	apply()
delattr()	help()	next()	setattr()	buffer()
dict()	hex()	object()	slice()	coerce()
dir()	id()	oct()	sorted()	intern()

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□

```
>>> help(raw_input)
```

□□□□□□

```
Help on built-in function raw_input in module __builtin__:

raw_input(...)
    raw_input([prompt]) -> string

    Read a string from standard input.  The trailing newline is
stripped.
    If the user hits EOF (Unix: Ctl-D, Windows: Ctl-Z+Return),
raise EOFError.
    On Unix, GNU readline is used if enabled.  The prompt
string, if given,
    is printed without a trailing newline before reading.
```

□□□□□□□□□□□□□ `raw_input()` □□□□□□□

□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□
https://docs.Python.org/2/library/functions.html

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ `print()` □□□□□

```
 print(*objects, sep=' ', end='\n', file=sys.stdout)

    Print objects to the stream file, separated by sep and
followed by end. sep, end and file, if present, must be given as
keyword arguments.

    All non-keyword arguments are converted to strings like
str() does and written to the stream, separated by sep and
```

```
followed by end. Both sep and end must be strings; they can also
be None, which means to use the default values. If no objects
are given, print() will just write end.

    The file argument must be an object with a write(string)
method; if it is not present or None, sys.stdout will be used.
Output buffering is determined by file. Use file.flush() to
ensure, for instance, immediate appearance on a screen.
```

□□□□□□□□□□□□□□□□□□□□□

```
>>> raw input("input your name:")
input your name:python
'python'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> name = raw input("input your name:")
input your name:python
>>> name
'python'
>>> type(name)
<type 'str'>
```

□□□□□□□□□ str □□□□□□□□□□□□□

```
>>> age = raw input("How old are you?")
How old are you?10
>>> age
'10'
>>> type(age)
<type 'str'>
```

□□□□□□□□□ str □□□

□□□ `print()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> print("hello, world")
hello, world
>>> a = "python"
>>> b = "good"
>>> print a
python
>>> print a,b
python good
```

□□□□□□□□□□□□□□□□□□□

[illegible]`print()`[illegible] `\n` [illegible] `\n`[illegible]

[illegible]“[illegible]”[illegible]

```
#!/usr/bin/env python
# coding=utf-8

name = raw_input("What is your name?")
age = raw_input("How old are you?")

print "Your name is:", name
print "You are " + age + " years old."

after_ten = int(age) + 10
print "You will be " + str(after_ten) + " years old after ten 
years."
```

[illegible]

[illegible] `print()` [illegible]

```
print "You are " + age + " years old."
```

[illegible] `age` [illegible]

```
print "You will be " + str(after_ten) + " years old after ten 
years."
```

[illegible] `after_ten` [illegible] str [illegible]

[illegible] `after_ten = int(age) + 10` [illegible] `raw_input` [illegible] str [illegible] age [illegible] 10 [illegible] `int()`[illegible] 10 [illegible]

[illegible]

[illegible]

[illegible]

```
>>> print "I like \npython"
I like
python
```

[illegible]“[illegible]”[illegible] n [illegible] DOS [illegible]DOS[illegible]

```
 >>> dos = "c:\news"
>>> dos
'c:\news'        # □□□□□□□□□□
>>> print dos    # □□ print □□□□□□□□□□□□□□□□□□
c:
ews
```

□□□□□□□□□□□□□□□□□□□

```
 >>> dos = "c:\\news"
>>> print dos
c:\news
```

□□□□□□□□□□□□

```
 >>> dos = r"c:\news"
>>> print dos
c:\news
>>> print r"c:\news\python"
c:\news\python
```

□□ `r"c:\news"` □□ r □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

[illegible](3)

[illegible]

[illegible] `Python` [illegible]P,y,t,h,o,n[illegible]

[illegible] Python [illegible]“[illegible]”[illegible]

[illegible] 108 [illegible]“[illegible]”[illegible] 108 [illegible]1 [illegible]2 [illegible]14 [illegible]22 [illegible]

[illegible] Python [illegible]([illegible] Python [illegible])[illegible]

[illegible]

[illegible] Python [illegible]

```
>>> lang = "study Python"
>>> lang[0]
's'
>>> lang[1]
't'
```

[illegible] lang[illegible] `s` [illegible] `lang[0]` [illegible] lang [illegible]

```
>>> "study Python"[0]
's'
```

[illegible] lang [illegible] `"study Python"` [illegible] Python [illegible] `lang[0]` [illegible] lang [illegible]

[illegible]1[illegible] 0 [illegible] Python[illegible] 0 [illegible] google [illegible]

0	1	2	3	4	5	6	7	8	9	10	11
s	t	u	d	y	l	p	y	t	h	o	n

[illegible]

[illegible]

```
>>> lang.index("p")
6
```

[illegible] 0[illegible]

[illegible]“[illegible]”[illegible]“[illegible] 29 [illegible] 34 [illegible]”

[illegible] lang[1] [illegible] t[illegible]“[illegible]”[illegible]“[illegible]”[illegible] t [illegible]

[illegible]“[illegible]”[illegible] Python [illegible]

```
>>> lang
'study Python'     #[illegible]“[illegible]”[illegible]
>>> lang[2:9]
'udy pyt'
```

[illegible] `lang[2:9]` [illegible] `2,3,4,5,6,7,8`([illegible])[illegible]

[illegible]

[illegible]“[illegible]”[illegible]

```
>>> lang
'study Python'
>>> b = lang[1:]    # [illegible] 1 [illegible]
>>> b
'tudy Python'
>>> c = lang[:]    # [illegible]
>>> c
'study Python'
>>> d = lang[:10]    # [illegible] 10 [illegible]
>>> d
'study pyth'
```

[illegible]

`lang[:10]`□□□□ `lang[0:10]`□□□□□

```
>>> e = lang[0:10]
>>> e
'study pyth'
```

□□□`lang[1:]`□ `lang[1:11]`□□□□□□□□□□□□□

```
>>> lang[1:11]
'tudy pytho'
>>> lang[1:]
'tudy python'
```

□□□`lang[1:12]`□□□□ 12 □□□□□□ 12 □□□□□□□□□□ 1 □ 11 □□□□□□□□

```
>>> lang[1:12]
'tudy python'
>>> lang[1:13]
'tudy python'
```

□□□□□□□□□□□□ 12□□ 13□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□ `c = lang[:]`□□□□□□□ c □□□□□□□□□□□□□“□□”□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> id(c)
3071934536L
>>> id(lang)
3071934536L
```

`id()`□□ c □ lang □□□□□□□□□□□□□□□ `c=lang[:]`□□□□□□□□□□□□□□□□□□□□□□ c □□□□□□□□□□□□□□□□□□

```
>>> lang = "study python"
>>> c = lang
```

□□□□□□□□□ c □ lang □□

□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

1. len()□□□□□□
2.

    - □□□ 2 □□□

3.

    - : □□□□□□

4. in :□□□□□□□□□□□□
5. max() :□□□□□
6. min() :□□□□□
7. cmp(str1,str2) :□□ 2 □□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

“+”□□□□□

```
>>> str1 + str2
'abcdabcde'
>>> str1 + "-->" + str2
'abcd-->abcde'
```

□□□
□□□□□

in

```
>>> "a" in str1
True
>>> "de" in str1
False
>>> "de" in str2
True
```

`in` □□□ `True` □□□□□
`False` □

□□

```
>>> max(str1)
'd'
>>> max(str2)
'e'
```

```
>>> min(str1)
'a'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`min()`□ `max()`□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ google □□□□□□□□□ ASCII □□□□□□□□□□□□

□□

```
>>> cmp(str1, str2)
-1
```

□□□ 0□ □□□□□□□□ 0□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> ord('a')
97
>>> ord('b')
98
>>> ord(' ')
32
```

`ord()`□□ ASCII □□□□□□□□□□□ a □ ASCII □□□□ 97□□□□ ASCII □□□□□□ 32□□□□□□□□□□□□□□□□□□□□□□□□□□□ `chr()`□

```
>>> chr(97)
'a'
>>> chr(98)
'b'
```

□□□□□□□□□□□□□□□

```
>>> cmp("a","b")    #a-->97, b-->98, 97 □□ 98□□□ a □□ b
-1
>>> cmp("abc","aaa")
1
>>> cmp("a","a")
0
```

□□□□□□□□□□□□□□□□□

```
>>> cmp("ad","c")
-1
```

□□ 0□□□□□□□□□□□□□□□□□□□□□ 0□0 □ ASCII □□□□□ NUL□□□□□□□□□□□□□□ad □□ a □□□

□□ c □□□□□□□ a □□ c□□□□□□□□ -1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□

```
 >>> cmp("123","23")
-1
>>> cmp(123,23)    # □□□□□□□□□□□□□□□□□□□□□□
1
```

“*”

□□□□□“□□”□□

```
 >>> str1*3
'abcdabcdabcd'
>>> print "-"*20    # □□□□□□□`-`
--------------------
```

len()

□□

> □□□
> □□□□□

□□□□□□□□□□□□

```
 >>> a="hello"
>>> len(a)
5
```

□□□□□□□□ len(object)□□□□□□□□□□□□□□□

```
 >>> m = len(a)  # □□□□□□□□□□□□□
>>> m
5
>>> type(m)     # □□□□□□□□□□□□□□□
<type 'int'>
```

[illegible]

[illegible]qiwsir@126.com,**[illegible]**

[illegible](4)

[illegible]

[illegible]

[illegible]

> [illegible]partition[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

> [illegible] C[illegible]C++ [illegible] printf [illegible]（conversion specification[illegible] 0 [illegible]

[illegible] C [illegible] Python [illegible] `print` [illegible]

[illegible]

```
>>> "I like %s"
'I like %s'
```

[illegible]`%s`[illegible]

```
>>> "I like %s" % "python"
'I like python'
>>> "I like %s" % "Pascal"
'I like Pascal'
```

[illegible]

[illegible] `%s` [illegible] `%d` [illegible] `%f` [illegible]

[illegible]	[illegible]

□□□	□□
%s	□□□(□□ str()□□□)
%r	□□□(□□ repr()□□□)
%c	□□□□
%b	□□□□□
%d	□□□□□
%i	□□□□□
%o	□□□□□
%x	□□□□□□
%e	□□ (□□□□ e)
%E	□□ (□□□□ E)
%f	□□□
%F	□□□□□□□□
%g	□□(e) □□□□ (□□□□□□)
%G	□□(E)□□□□ (□□□□□□)

□□□□

```
 >>> a = "%d years" % 15
>>> print a
15 years
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> print "Suzhou is more than %d years. %s lives in here." % 
(2500, "qiwsir")
Suzhou is more than 2500 years. qiwsir lives in here.
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> print "Today's temperature is %.2f" % 12.235
Today's temperature is 12.23
>>> print "Today's temperature is %+.2f" % 12.235
Today's temperature is +12.23
```

[illegible]

[illegible]

[illegible]“[illegible]”[illegible] Python [illegible]

```
>>> s1 = "I like {}".format("python")
>>> s1
'I like python'
>>> s2 = "Suzhou is more than {} years. {} lives in
here.".format(2500, "qiwsir")
>>> s2
'Suzhou is more than 2500 years. qiwsir lives in here.'
```

[illegible] Python [illegible] `string.format()` [illegible] `{}` [illegible]

[illegible] format [illegible] `{}` [illegible]

[illegible]

```
>>> print "Suzhou is more than {year} years. {name} lives in
here.".format(year=2500, name="qiwsir")
Suzhou is more than 2500 years. qiwsir lives in here.
```

[illegible]

[illegible]“[illegible]”[illegible]

```
>>> lang = "Python"
>>> print "I love %(program)s"%{"program":lang}
I love Python
```

[illegible]`string.format()`

[illegible]

[illegible] dir [illegible]

```
>>> dir(str)
['  add  ', '  class  ', '  contains  ', '  delattr  ',
'  doc  ', '  eq  ', '  format  ', '  ge  ', '  getattribute__',
'  getitem  ', '  getnewargs  ', '  getslice  ', '  gt _',
'  hash  ', '  init  ', '  le  ', '  len  ', '  lt  ',
'  mod  ', '  mul  ', '  ne  ', '  new  ', '  reduce _',
'  reduce ex  ', '  repr  ', '  rmod  ', '  rmul  ',
'  setattr  ', '  sizeof  ', '  str  ', '  subclasshook__',
' formatter field name split', ' formatter parser',
'capitalize', 'center', 'count', 'decode', 'encode', 'endswith',
'expandtabs', 'find', 'format', 'index', 'isalnum', 'isalpha',
'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join',
'ljust', 'lower', 'lstrip', 'partition', 'replace', 'rfind',
'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split',
'splitlines', 'startswith', 'strip', 'swapcase', 'title',
'translate', 'upper', 'zfill']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ help □□□□□□

```
>>> help(str.isalpha)

Help on method_descriptor:

isalpha(...)
    S.isalpha() -> bool

    Return True if all characters in S are alphabetic
    and there is at least one character in S, False otherwise.
```

□□□□□□□□□□□□□□□□□□□□□□

```
>>> "python".isalpha()    # □□□□□□□□□□□□ True
True
>>> "2python".isalpha()    # □□□□□□□□□□ False
False
```

split

□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> a = "I LOVE PYTHON"
>>> a.split(" ")
['I', 'LOVE', 'PYTHON']
```

□□□□□□□□□□□□□□□□□□□□□□list□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> b = "www.itdiffer.com"
>>> b.split(".")
['www', 'itdiffer', 'com']
```

□□□□□□□□□□

[illegible]

[illegible]Python [illegible]

[illegible]

- S.strip() [illegible]
- S.lstrip() [illegible]
- S.rstrip() [illegible]

[illegible]

```
>>> b=" hello "    # [illegible]
>>> b.strip()
'hello'
>>> b
' hello '
```

[illegible]

```
>>> b.lstrip()    # [illegible]
'hello '
>>> b.rstrip()    # [illegible]
' hello'
```

[illegible]

[illegible][[illegible]
href="https://github.com/qiwsir/algorithm/blob/master/string_to_hump.md")[illegible]

[illegible] Python [illegible]

- S.upper() #S [illegible]
- S.lower() #S [illegible]
- S.capitalize() # [illegible]
- S.isupper() #S [illegible]
- S.islower() #S [illegible]
- S.istitle()

[illegible]

```
>>> a = "qiwsir,Python"
>>> a.upper()       # □□□□□□□□□□□□□
'QIWSIR,PYTHON'
>>> a               # □□□□□□□□□□
'qiwsir,Python'
>>> b = a.upper()
>>> b
'QIWSIR,PYTHON'
>>> c = b.lower()   # □□□□□□□□□□□□□□
>>> c
'qiwsir,Python'

>>> a
'qiwsir,Python'
>>> a.capitalize()  # □□□□□□□□□□□□□□
'Qiwsir,Python'
>>> a               # □□□□□□□□□
'qiwsir,Python'
>>> b = a.capitalize() # □□□□□□
>>> b
'Qiwsir,Python'

>>> a = "qiwsir,github"    # □□□□□□□□□□□□□□□□□□□□□□
>>> a.istitle()
False
>>> a = "QIWSIR"        # □□□□□□□□□□□ False
>>> a.istitle()
False
>>> a = "qIWSIR"
>>> a.istitle()
False
>>> a = "Qiwsir,github"  # □□□□□□□□ False
>>> a.istitle()
False
>>> a = "Qiwsir"        # □□□ True
>>> a.istitle()
True
>>> a = 'Qiwsir,Github' # □□□□ True
>>> a.istitle()
True

>>> a = "Qiwsir"
>>> a.isupper()
False
>>> a.upper().isupper()
True
```

```
>>> a.islower()
False
>>> a.lower().islower()
True
```

[illegible]

```
 >>> a = "This is a Book"
>>> a.istitle()
False
>>> b = a.title()     # [illegible]
>>> b
'This Is A Book'
>>> b.istitle()       # [illegible]
True
```

join [illegible]

[illegible]“+”[illegible]“+”[illegible]

[illegible] join [illegible]

```
 >>> b
'www.itdiffer.com'
>>> c = b.split(".")
>>> c
['www', 'itdiffer', 'com']
>>> ".".join(c)
'www.itdiffer.com'
>>> "*".join(c)
'www*itdiffer*com'
```

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□□□

□□□□□□□□□□□ Python□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□“□”□

□□□□□□ Python2 □□□□□□□□□□□□□□□□Python3 □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ Python2□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ str□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> name = '□□'
>>> name
'\xe8\x80\x81\xe9\xbd\x90'
```

□□

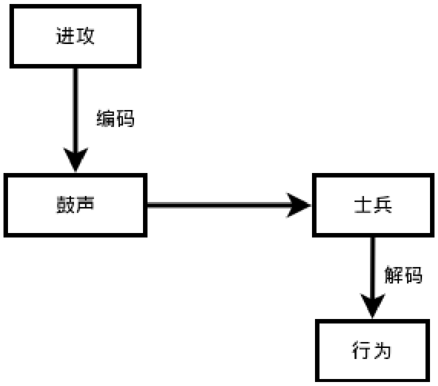
```
>>> name = "qiwsir"
>>> name
'qiwsir'
```

□□

□□

□□

□□“□□”□□□□□□□□□□□


进攻
编码
鼓声
士兵
解码
行为

1. □□

2. [illegible]“[illegible]”[illegible]
3. [illegible]

[illegible]

[illegible]

> [illegible] 19 [illegible] 20 [illegible]
>
> [illegible] 1871 [illegible] 6 [illegible] 3 [illegible]1877 [illegible] 10 [illegible]1879 [illegible]1880 [illegible] 1881 [illegible] 12 [illegible]“[illegible]”[illegible]1887 [illegible]1884 [illegible]"[illegible]"[illegible] 8 [illegible] 5 [illegible]8 [illegible] 22 [illegible] 8 [illegible] 30 [illegible]
>
> [illegible] 4 [illegible] 3 [illegible]

[illegible]“[illegible]”[illegible]

> 1873 [illegible],[illegible],[illegible] 6800 [illegible],[illegible]

[illegible] Morse Code

> [illegible]Morse Code[illegible]·[illegible] 1836 [illegible]
>
> [illegible] 0 [illegible] 1 [illegible].[illegible]-[illegible]

[illegible]

> [illegible] 1999 [illegible]1997 [illegible]“[illegible]”

```
****-/*----/----*/****-/****-/*----/---**/*----/****-/*----/-****/***-
```

```
****-/*----/----*/****-/****-/*----/---**/*----/****-/*----/-****/***-
```

□□□□□□□□□□□□□□□□

□□□□□□□□□□□

□□□□□□□

□□□□□□□□□□□□□

> □□□□□□□□Character encoding□□8 □□ ASCII□□□□ASCII □□□□□□□□□□□□□□□□ 7 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1 □□□□□□□□□
>
> □□□□□□□□□□□□□ ASCII□1963 □□□ EBCDIC□1964 □□ CJK □□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ASCII □□□

> ASCII□pronunciation: □□□□□/'æski/ ASS-kee[1]□American Standard Code for Information Interchange□□□ EASCII □□□□□□□□□□□□□□□□□□□□□□□□ ISO/IEC 646□□□□□□□□ ASCII □□□□□□□2007 □ 12 □□□□□□ Unicode □□□

□□□□□□□□□□□□□□□□□□□□ Unicode □□□□□ Unicode □□□□□□□□□□□□□

> Unicode□□
>
> Unicode □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Unicode □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 7.0.0□□□□□□□□□□□□□□□□□□□□□ 2005 □□□□□□Unicode □□

□□□□□□□□□□□□□□□□□□□□□□□□□ Unicode □□□□□....□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Unicode □□□□□□ Unicode □□□□Unicode Transformation Format□□□ UTF□□□□□□□□□□□□□□□□ utf-8□

□□□ utf-8□□□□□□□□□□□

> UTF-8□8-bit Unicode Transformation Format□□□□□□ Unicode □□□□□□□□□□□□□ □□□□□□□□□□□□ Unicode □□□□□□□□□□□□□□□□□□□□□□ ASCII □□□□□□□□□□ ASCII □ □□

□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□coding:utf-8 □□□□□□□□□ python □□□□□□□□□□ □□

encode □ decode

□□□ encode()□ decode()□□□□□ □□□□

> codecs.encode(obj[, encoding[, errors]]):Encodes obj using the codec registered for encoding. codecs.decode(obj[, encoding[, errors]]):Decodes obj using the codec registered for encoding.

Python2 □□□□□□ ascii□□□ encode □□□□□□□□□□□□□□□□□□□“□□”□□□ decode □□□ □□□□□□□□□□“□□”□□

□□□□□□□□□□□

```
>>> a = "□"
>>> type(a)
<type 'str'>
>>> a
'\xe4\xb8\xad'
>>> len(a)
3

>>> b = a.decode()
>>> b
u'\u4e2d'
>>> type(b)
<type 'unicode'>
>>> len(b)
1
```

□□ □□

□□□□□□□□ a □□□□□□□□□□□□□□□□□□(str)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□“□”□□□□□□□□□□□□□□□□□□□□□□□□□□□ google □□□□□ len(a)□□□□□□□□□□ □□□□□□□□

[illegible] decode [illegible] unicode [illegible] unicode [illegible] 1.

[illegible] unicode [illegible]

```
>>> c = b.encode('utf-8')
>>> c
'\xe4\xb8\xad'
>>> type(c)
<type 'str'>
>>> c == a
True
```

[illegible] google[illegible]

Python [illegible]

[illegible]

[illegible] utf-8 [illegible]

[illegible]

```
# -*- coding: utf-8 -*-
```

[illegible]-*-[illegible]

```
# coding:utf-8
```

[illegible] unicode[illegible] str()[illegible] unicode()

```
unicode_str = unicode('□□', encoding='utf-8')
print unicode_str.encode('utf-8')
```

[illegible] codecs.open[illegible] open([illegible])

```
import codecs
codecs.open('filename', encoding='utf8')
```

[illegible][Python2.x[illegible] href="https://github.com/qiwsir/ITArticles/blob/master/Python/Python%E7%9A%84%E4%B8%AD%E6%96%87%E6%98%BE%E7%A4%BA%E6%96%B9%E6%B3%95.md")

[illegible] Python3[illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

□□(1)

□□□□□□□□□□□□□□□□ Python □□□□□□int □ str□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□ Python □□□□□□□□□□□□□□ □ 2,3 □□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□ int □□□□□□□□□ str □□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ int □□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□ □□□□□□□□□□□□ JAVA □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ list □□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□

LIST □ Python □□□□□□□□□□□

□□

□ Python □□□□□□□□□□ list□[ ]

□□□□□□□□□□ int□□□□□ str □□□□□□□□□□□□ True/False □□□□□□□□□□□□□□□□□□□□□□

```
>>> a=[]        #□□□□□□□ a□□□ list □□□□□□□□□
>>> type(a)
<type 'list'>   #□□□□□ type()□□□□ a □□□□□ list
>>> bool(a)     #□□□□□ bool()□□ list □□□□□ a □□□□□□□□□□□□□□
False
False
>>> print a     #□□ list □□□□□ a
[]
```

`bool()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□”□□“□”□□□□□□□□□□□□□□□□list □ □□□□□□□□□□□□ bool()□□□□□□□□ False□□□□□□□□□□

□□□□□□□□□□□□□

```
>>> a=['2',3,'qiwsir.github.io']
>>> a
['2', 3, 'qiwsir.github.io']
>>> type(a)
<type 'list'>
```

```
>>> bool(a)
True
>>> print a
['2', 3, 'qiwsir.github.io']
```

□□□□□□□□□□□ list □□□□□□□□□□

□□□□□□□“□□□□□□ list”□□□□□□ list □□□□□□□□□□□□□□□□□ list □□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□

> □□□□□□□□□□□□□□□□□□□□□□□□□□□□ Java□□□□□□□□ list □□□□□□□□——□□——□□□□□□□□□ □□□□□ Java □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ int □□□□□□□□ char □□□□□□□□□□□□ □□ int □□□□ char □□□□□□□□ java □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ python □□ list□□□□ java □□□□□□□□□□□——□□ `[ ]` □□□——list □□□□□□□□□□□□□□□□ int,str□□□ □□□□□ list□□□□□□□□□□□ dict □□□□□□□□□□□□List □ python □□□□□□□□□□□□□

□□□□□

□□□□□□□□(3)□□□□□□“□□”(index)□“□□”□

```
>>> url = "qiwsir.github.io"
>>> url[2]
'w'
>>> url[:4]
'qiws'
>>> url[3:9]
'sir.gi'
```

□ list □□

```
>>> a
['2', 3, 'qiwsir.github.io']
>>> a[0]    #□□□□□□□ 0 □□
'2'
>>> a[1]
3
>>> [2]
[2]
>>> a[:2]   #□str□□□□□□□□□□□□□□□□□□□□□□□□
['2', 3]    #□□□□□□□
>>> a[1:]
[3, 'qiwsir.github.io']
```

list □ str □□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> lang = "python"
>>> lang.index("y")
1
>>> lst = ['python','java','c++']
>>> lst.index('java')
1
```

□□ 0□□□□□□□ 1□□□□□□□□□□□□□□□□□□□□□□ `-1` □□□□□□□□□-2,-3,...□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> lang
'python'
>>> lang[-1]
'n'
>>> lst
['python', 'java', 'c++']
>>> lst[-1]
'c++'
```

□□□□□□□□□ -1 □□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> lang[-1:-3]
''
>>> lang[-3:-1]
'ho'
>>> lst[-3:-1]
['python', 'java']
```

□□□□□□□□□□□□□□□□□□□□□□□`lang[-1:-3]`□□□□□□□□□□□□□□□□□□□

□□

□□□

```
>>> alst = [1,2,3,4,5,6]
>>> alst[::-1]    #□□
[6, 5, 4, 3, 2, 1]
>>> alst
[1, 2, 3, 4, 5, 6]
```

□□□□□□□□□□□

```
>>> lang
'python'
>>> lang[::-1]
```

```
'nohtyp'
>>> lang
'python'
```

[illegible] str [illegible] lst [illegible]“[illegible]”[illegible]

[illegible]Python [illegible]list[illegible]

```
>>> list(reversed(alst))
[6, 5, 4, 3, 2, 1]
```

[illegible]

[illegible] reversed [illegible]

```
>>> help(reversed)
Help on class reversed in module __builtin__:

class reversed(object)
 |  reversed(sequence) -> reverse iterator over values of the 
sequence
 |
 |  Return a reverse iterator
```

[illegible]

```
>>> list(reversed("abcd"))
['d', 'c', 'b', 'a']
```

[illegible]

[illegible] list [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] Python [illegible]

[illegible] str [illegible] s.strip()[illegible] str [illegible] list [illegible]

[illegible]list [illegible]

□□□□

- len()

□□□□□□□□□

```
>>> lst
['python', 'java', 'c++']
>>> len(lst)
3
```

- +□□□□□□□

□□□□□□

```
>>> lst
['python', 'java', 'c++']
>>> alst
[1, 2, 3, 4, 5, 6]
>>> lst + alst
['python', 'java', 'c++', 1, 2, 3, 4, 5, 6]
```

- *□□□□□

□□□□□□□

```
>>> lst
['python', 'java', 'c++']
>>> lst * 3
['python', 'java', 'c++', 'python', 'java', 'c++', 'python',
'java', 'c++']
```

- in

□□ lst □□□□□□

```
>>> "python" in lst
True
>>> "c#" in lst
False
```

- max()□ min()

□ int □□□□□□□□□□□□□□□□□□□ ascii □□□□□□□□□□□□□□□

```
>>> alst
[1, 2, 3, 4, 5, 6]
>>> max(alst)
```

```
6
>>> min(alst)
1
>>> max(lst)
'python'
>>> min(lst)
'c++'
```

- cmp()

[illegible]

```
 >>> lsta = [2,3]
>>> lstb = [2,4]
>>> cmp(lsta,lstb)
-1
>>> lstc = [2]
>>> cmp(lsta,lstc)
1
>>> lstd = ['2','3']
>>> cmp(lsta,lstd)
-1
```

[illegible]

```
 >>> a = ["good","python","I"]
>>> a
['good', 'python', 'I']
>>> a.append("like")        #[illegible] list [illegible] str [illegible] "like"
>>> a
['good', 'python', 'I', 'like']
>>> a.append(100)           #[illegible] list [illegible] int [illegible] 100
>>> a
['good', 'python', 'I', 'like', 100]
```

[illegible] list.append()[illegible]

> list.append(x)
>
> Add an item to the end of the list; equivalent to a[len(a):] = [x].

[illegible]“[illegible]”[illegible] list.append(x)[illegible] x [illegible] list [illegible]

[illegible] equivalent to a[len(a):] = [x][illegible] list.append(x)[illegible]a[len(a):]=[x][illegible]

```
>>> a
['good', 'python', 'I', 'like', 100]
>>> a[len(a):]=[3]      #len(a),□□□ list □□□□□□□□□□ list □□□□□□□
>>> a
['good', 'python', 'I', 'like', 100, 3]
>>> len(a)
6
>>> a[6:]=['xxoo']
>>> a
['good', 'python', 'I', 'like', 100, 3, 'xxoo']
```

□□□

□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

□□(2)

□□□□□□□□□list □ Python □□
□□□□□

```
>>> dir(list)
['  add  ', '  class  ', '  contains  ', '  delattr _',
'  delitem  ', '  delslice  ', '  doc  ', '  eq  ',
'  format  ', '  ge  ', '  getattribute  ', '  getitem  ',
'  getslice  ', '  gt  ', '  hash  ', '  iadd  ', '  imul__',
'  init  ', '  iter  ', '  le  ', '  len  ', '  lt  ',
'  mul  ', '  ne  ', '  new  ', '  reduce  ', '  reduce_ex__',
'  repr  ', '  reversed  ', '  rmul  ', '  setattr  ',
'  setitem  ', '  setslice  ', '  sizeof  ', '  str  ',
'  subclasshook  ', 'append', 'count', 'extend', 'index',
'insert', 'pop', 'remove', 'reverse', 'sort']
```

□□□□□□□□□□□□□□□□□□□□□□□□`__add__`□□□□□□□□□□□□□□□□□

> 'append', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort'

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

list □□

append □ extend

□□□(1)□□□□□□ list □□□□□□□□□ list.append(x)□□□□□□□□□□ x □□□□□□□□□□□ list □□□

□□□□□□□□□□□ list □□□□□□□□ list □□□□□□□□□□□□ list □□□□□□□□□□ list □□□□□□□□□□□□□□□□□□□□
□□□□□

> list.extend(L)
>
> Extend the list by appending all the items in the given list; equivalent to a[len(a):] = L.

□□

□□□□□□□□□□□□□

> □□□□□□□□□□□□□ list □□□□□□□□ a[len(a):]= L

□□□□□□□□□□□□□□□□□□□□□

```
>>> la
[1, 2, 3]
>>> lb
['qiwsir', 'python']
>>> la.extend(lb)
>>> la
[1, 2, 3, 'qiwsir', 'python']
>>> lb
['qiwsir', 'python']
```

□□□□□□□□□□□□□□□ list□□□□ la□□□□□ lb□□ lb □□□ la □□□□□□□□□ lb □□□□□□□□□□ la □□ □□ la □□□

□□

```
>>> la = [1,2,3]
>>> b = "abc"
>>> la.extend(b)
>>> la
[1, 2, 3, 'a', 'b', 'c']
>>> c = 5
>>> la.extend(c)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  TypeError: 'int' object is not iterable
```

□□□□□□□□□□□□□□□□□□□□□□□□□ extend(str)□□□□str □□□□□□□□□□□□□□□□□ la □□□

□□ extend □□□□□□□□□□□□□

□□□extend □□□□□□ list□□□□ str□□ Python □□□□□□□□□□□□□□□□ list □□□□□□ list□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> la
[1, 2, 3, 'a', 'b', 'c']
>>> lb
['qiwsir', 'python']
>>> la[len(la):]=lb
>>> la
[1, 2, 3, 'a', 'b', 'c', 'qiwsir', 'python']
```

list.extend(L) □□□ list[len(list):] = L□L□□□□□ list

□□□□□□□□□□□ list □□ list.append(),□□□□□□

> extend(...) L.extend(iterable) -- extend list by appending elements from the iterable

□□□□□□□□□□□ `help(list.extend)` □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□extend □□□□□□□□□□□□□□□□□□□□□□□□□□□ iterable□□□□ iterable□□□□□
□□□□□□□□□□□□□□□□□□□□□□

> iterable,□□□□□“□□□□”□□ Python □□□□□□□□□□ iterator□□□□□“□□□”□□□□□□□□□□
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ iterable(□□□□)□□□□□□□□“□□”□□□□□□□□

> □□“□□”□□□□□□□□□□"light amplification by stimulated emission of radiation"□□□□"laser"□□□“□”□□□□□□□□□□□□“□”□□□□□□□□“□”□□□□□“□□”□□□□□□□□□“□□”□□□□□□□□□□
>
> “□”□□□□□□□□“□□,□□”□□:□□□□□□□□“□”□□□□□□□□“□□‘□’”□□□□□□“□□□□”□□□□□“□□-□□-□□□-□□-□□-□□□-…”□□□□□□□□□“□□”□□□□□□
>
> □□□□□□□□□□□□□□□□□“□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□”

□□□□□□□□“□□□”(iterable)□□□

```
>>> astr = "Python"
>>> hasattr(astr,'__iter__')
False
```

□□□□□□□ `hasattr()` □□□□□□□□□□□□□□□□□□ False□□□□□□□□□□□□

```
>>> alst = [1,2]
>>> hasattr(alst,'__iter__')
True
>>> hasattr(3, '__iter__')
False
```

`hasattr()` □□□□□□□□□□□□□□□□`__iter__`□□□□□□□□ `dir()` □□□□□□□□□□□□□□□□□□□`__iter__`□□□□□□□□ dict,tuple □□□□□□□□□□□□□□□

□□□□□□□□□□□“iterable”□□□□□□□□□□□ extend □□□□□□□□□□□□□ iterable □□□□□□

```
>>> new = [1,2,3]
>>> lst = ['Python','qiwsir']
>>> lst.extend(new)
>>> lst
['Python', 'qiwsir', 1, 2, 3]
>>> new
[1, 2, 3]
```

□□ extend □□□□[1,2,3]□□□□□□□□□□□□□□□ lst □□

□□□□□□□□□□□□□□□□□□□□lst □□ extend □□□□□□□□□□□□□□□“□”□□□□□□□□□□□□□□“□□□”□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□

```
>>> new = [1,2,3]
>>> id(new)
3072383244L

>>> lst = ['python', 'qiwsir']
>>> id(lst)
3069501420L
```

□ `id()` □□□□□□□□□□□□□□□□“□”□□□□

```
>>> lst.extend(new)
>>> lst
['python', 'qiwsir', 1, 2, 3]
>>> id(lst)
3069501420L
```

□□□□□□□□□□ lst □□ `extend()` □□□□□□□□□□□□□□□□□□□□□□□□“□”□□□□□□□□□□□□“□”□□“□”□□□“□”□□□□□□□□□□□□□□□□

□□

□□□`append()`□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□ lst □□□□□□□□□□□□□□□□□ shell □□□□□□ id □□□□□□□□□□□□□“□”□□□□□□

```
>>> lst = ['Python','qiwsir']
>>> id(lst)
3069501388L
>>> lst.append(new)
>>> lst
['Python', 'qiwsir', [1, 2, 3]]
>>> id(lst)
3069501388L
```

□□□`append()`□□□□□□□□□□

□□□□□ `extend()` □□□□□□ iterable □□□□□□□□□□

```
>>> lst.extend("itdiffer")
>>> lst
['python', 'qiwsir', 'i', 't', 'd', 'i', 'f', 'f', 'e', 'r']
```

□□□□□□□"itdiffer"□□□['i', 't', 'd', 'i', 'f', 'f', 'e', 'r']□□□□□□□□□□□□□□□□ extend□□□□
□□□□□□□□□□□□□□

```
>>> num_lst = [1,2,3]
>>> num_lst.extend(8)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'int' object is not iterable
```

□□□□□□□□□□□□□□□□□□□□ 8□□ int □□□□□□□□ iterable □□

□□□□□□□□□□□□□□□ `append()` □ `extend()` □□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□
- □□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

```
>>> one = ["good","good","study"]
>>> another = one.extend(["day","day","up"])    #□□□□□□□□□□□□□□□
□□□□□□□□
>>> another                                     #□□□□□□□□□□□□
>>> one
['good', 'good', 'study', 'day', 'day', 'up']
```

□□□□□□□□□□□□□□□□□□

```
>>> lst = [1,2,3]
>>> lst.append(["qiwsir","github"])
>>> lst
[1, 2, 3, ['qiwsir', 'github']]  #append □□□
>>> len(lst)
4

>>> lst2 = [1,2,3]
>>> lst2.extend(["qiwsir","github"])
>>> lst2
[1, 2, 3, 'qiwsir', 'github']   #extend □□□
>>> len(lst2)
5
```

append □□□□□□□□extend □□□□□□□

count

□□□ len(L)□□□□ list □□□□□□□ list □□□□□□□□python □ list □□□□□□□□□□□□□□□□□□ list □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> list.count(x)
>
> Return the number of times x appears in the list.

□□□□□□□□□□□□□□□□□□□□□

```
>>> la = [1,2,1,1,3]
>>> la.count(1)
3
>>> la.append('a')
>>> la.append('a')
>>> la
[1, 2, 1, 1, 3, 'a', 'a']
>>> la.count('a')
2
>>> la.count(2)
1
>>> la.count(5)     #NOTE:la □□□ 5,□□□□□□□□□□□□□□□□□□□□□ 0
0
```

index

□□□(1)□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> la
[1, 2, 3, 'a', 'b', 'c', 'qiwsir', 'python']
>>> la.index(3)
2
>>> la.index('qi')      #□□□□□□□□□
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  ValueError: 'qi' is not in list
>>> la.index('qiwsir')
6
```

list.index(x)□x □ list □□□□□□□□□□□□□□□□□□□ list □□□□□□□□□□□□□□□□□□□□□□□ index□

□□□□□□□□□□□

> list.index(x)

> Return the index in the list of the first item whose value is x. It is an error if there is no such item.

[illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

□□(3)

□□□□□□□□□□□□□□□□□□□□□□□□□

> 'append', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort'

□□□□□□□□□□□□

□□□

list □□

insert

□□□□ list □□□□□□□□□□□□□□□□□□□□□□□□□□ list □□□□□□□□

```
>>> all users = ["qiwsir","github"]
>>> all users.append("io")
>>> all users
['qiwsir', 'github', 'io']
```

□ `list.append(x)` □□□ `list.insert(i,x)` □□□ list □□□□□□□□□□□□□□□□□□□□□□□
□□

□□□□□□□□□□□□

> list.insert(i, x)
>
> Insert an item at a given position. The first argument is the index of the element before which to insert, so a.insert(0, x) inserts at the front of the list, and a.insert(len(a), x) is equivalent to a.append(x).

□□...□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> all users
['qiwsir', 'github', 'io']
>>> all users.insert("python")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: insert() takes exactly 2 arguments (1 given)
```

□□□□□□□□□□□□`insert()`□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> all users.insert(0,"python")
>>> all users
['python', 'qiwsir', 'github', 'io']

>>> all users.insert(1,"http://")
>>> all users
['python', 'http://', 'qiwsir', 'github', 'io']
```

`list.insert(i, x)`□□ i □□□□ x □□□ list □□□□□□□ x □□□□□□□ i □□□□□□□□□□□□□
0 □□□□

□□□□□□□□□□□□□□□

```
 >>> length = len(all_users)
>>> length
5
>>> all users.insert(length,"algorithm")
>>> all users
['python', 'http://', 'qiwsir', 'github', 'io', 'algorithm']
```

□ all users □□□□□□□□□ 4□□□□ `all_users.insert(5,"algorithm")`□□□□
□`"algorithm"`□□□□□□□ 5 □□□□□□□□□□□□□□5 □□□□ 4 □□□□□□□□□□□□□

□□□□□□□□□

```
 >>> a = [1,2,3]
>>> a.insert(9,777)
>>> a
[1, 2, 3, 777]
```

□□□□□□□□□□□ i □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

pop □ remove

list □□□□□□□□□□□□□□□□□□□ list □□□□□□□□□□□□□□□

> list.remove(x)
>
> Remove the first item from the list whose value is x. It is an error if there is no such item.
>
> list.pop([i])

> Remove the item at the given position in the list, and return it. If no index is specified, a.pop() removes and returns the last item in the list. (The square brackets around the i in the method signature denote that the parameter is optional, not that you should type square brackets at that position. You will see this notation frequently in the Python Library Reference.)

[illegible] Python[illegible]

[illegible] list.remove(x)[illegible] list [illegible] x [illegible] list [illegible]

```
>>> all_users
['Python', 'http://', 'qiwsir', 'github', 'io', 'algorithm']
>>> all_users.remove("http://")
>>> all_users       #[illegible]"http://"[illegible]
['Python', 'qiwsir', 'github', 'io', 'algorithm']

>>> all_users.remove("tianchao")        #[illegible] list [illegible]“tianchao”[illegible]
[illegible]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: list.remove(x): x not in list

>>> lst = ["python","java","python","c"]
>>> lst.remove("python")
>>> lst
['java', 'python', 'c']
```

[illegible] lst[illegible] list,[illegible] list [illegible] Python [illegible]

[illegible]'Python'[illegible]'Python'[illegible]**remove the first item ...**

[illegible]

- [illegible]
- [illegible] list [illegible]x not in list

> [illegible] Python [illegible] Python [illegible]
>
> and, assert, break, class, continue, def, del, elif, else, except, exec, finally, for, from, global, if, import, in, is, lambda, not, or, pass, print, raise, return, try, while, with,yield
>
> [illegible]

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ list □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□Python □□□□□□□□

```
>>> all_users
['python', 'qiwsir', 'github', 'io', 'algorithm']
>>> "Python" in all_users       #□□□ in □□□□□□□□□□ list□□□□□□
True□□□□False
True

>>> if "Python" in all_users:
...     all_users.remove("python")
...     print all_users
... else:
...     print "'Python' is not in all_users"
...
['qiwsir', 'github', 'io', 'algorithm']     #□□□"Python"□□

>>> if "Python" in all_users:
...     all_users.remove("Python")
...     print all_users
... else:
...     print "'Python' is not in all_users"
...
'Python' is not in all_users        #□□□□□□□□□□□□□□□
```

□□□if-else□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□

□□□□□ list.pop([i])□□□□□□□□□□□□□□

```
>>> all_users
['qiwsir', 'github', 'io', 'algorithm']
>>> all_users.pop()     #list.pop([i]),□□□□□□[i]□□□□□□□□□□□
'algorithm'             #□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

>>> all_users
['qiwsir', 'github', 'io']

>>> all_users.pop(1)        #□□□□□□□ 1 □□□"github"
'github'

>>> all_users
['qiwsir', 'io']
>>> all_users.pop()
'io'
```

```
>>> all users           #[illegible] 0
['qiwsir']
>>> all users.pop(1)    #[illegible] 1 [illegible]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: pop index out of range      #[illegible] 1 [illegible] list [illegible]
[illegible]
```

[illegible] `list.remove(x)` [illegible] `list.pop([i])` [illegible] i [illegible] i [illegible]

[illegible] list [illegible] len(list)[illegible])[illegible]

reverse

reverse [illegible]

```
>>> a = [3,5,1,6]
>>> a.reverse()
>>> a
[6, 1, 5, 3]
```

[illegible] reversed[illegible]

> [illegible] `list.reverse()` [illegible] reversed [illegible]

sort

sort [illegible]

> sort(...)
>
> L.sort(cmp=None, key=None, reverse=False) -- stable sort *IN PLACE*; cmp(x, y) -> -1, 0, 1

```
>>> a = [6, 1, 5, 3]
>>> a.sort()
>>> a
[1, 3, 5, 6]
```

`list.sort()` [illegible]

```
>>> a.sort(reverse=True)
>>> a
[6, 5, 3, 1]
```

[illegible]

[illegible] key[illegible]

```
>>> lst = ["Python","java","c","pascal","basic"]
>>> lst.sort(key=len)
>>> lst
['c', 'java', 'basic', 'Python', 'pascal']
```

[illegible]

[illegible] sorted[illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□ list □ str

list □ str □□□
□“□□□□□”□

□□□

□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□”□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□ 0 □□□□□□□□ 1 □□□

□□□

```
>>> welcome str = "Welcome you"
>>> welcome_str[0]
'W'
>>> welcome_str[1]
'e'
>>> welcome_str[len(welcome_str)-1]
'u'
>>> welcome_str[:4]
'Welc'
>>> a = "python"
>>> a*3
'pythonpythonpython'

>>> git list = ["qiwsir","github","io"]
>>> git list[0]
'qiwsir'
>>> git_list[len(git_list)-1]
'io'
>>> git list[0:2]
['qiwsir', 'github']
>>> b = ['qiwsir']
>>> b*7
['qiwsir', 'qiwsir', 'qiwsir', 'qiwsir', 'qiwsir', 'qiwsir',
'qiwsir']
```

□□□□□□□□□□□□□□□□□□

```
>>> first = "hello,world"
>>> welcome_str
'Welcome you'
>>> first+","+welcome_str   #□ + □□□ str
'hello,world,Welcome you'
>>> welcome_str             #□□□ str □□□□□□□□□□□+□□□□□□□□□□□□□□
'Welcome you'
>>> first
'hello,world'

>>> language = ['python']
>>> git_list
['qiwsir', 'github', 'io']
>>> language + git_list     #□ + □□□ list□□□□□□□ list
['python', 'qiwsir', 'github', 'io']
>>> git_list
['qiwsir', 'github', 'io']
>>> language
['Python']

>>> len(welcome_str)    # □□□□□
11
>>> len(git_list)       # □□□□□
3
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

list □ str □□□□□□□list □□□□□□□str □□□□□□□□□□□□

□□□□ list □□□□□□□□□□□□□□ list □□□□□□

```
>>> git_list
['qiwsir', 'github', 'io']

>>> git_list.append("python")
>>> git_list
['qiwsir', 'github', 'io', 'python']

>>> git_list[1]
'github'
>>> git_list[1] = 'github.com'
>>> git_list
['qiwsir', 'github.com', 'io', 'python']

>>> git_list.insert(1,"algorithm")
>>> git_list
['qiwsir', 'algorithm', 'github.com', 'io', 'python']

>>> git_list.pop()
'python'

>>> del git_list[1]
>>> git_list
['qiwsir', 'github.com', 'io']
```

□□□□□□□□□□□ str □□□□□□□□□□

```
>>> welcome_str
'Welcome you'

>>> welcome_str[1]='E'
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item assignment

>>> del welcome_str[1]
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: 'str' object doesn't support item deletion

>>> welcome_str.append("E")
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
AttributeError: 'str' object has no attribute 'append'
```

□□□□□□□ str□□□□□□□

```
>>> welcome_str
'Welcome you'
>>> welcome_str[0]+"E"+welcome_str[2:]  #□□□□□□ str
'WElcome you'
>>> welcome_str                         #□□□□□□□□□□
'Welcome you'
```

□□□□□□□□□□□□□□□□□□□□ str□

□□ list

[illegible] str [illegible] list [illegible]

```
>>> matrix = [[1,2,3],[4,5,6],[7,8,9]]
>>> matrix = [[1,2,3],[4,5,6],[7,8,9]]
>>> matrix[0][1]
2
>>> mult = [[1,2,3],['a','b','c'],'d','e']
>>> mult
[[1, 2, 3], ['a', 'b', 'c'], 'd', 'e']
>>> mult[1][1]
'b'
>>> mult[2]
'd'
```

[illegible] list [illegible] list [illegible]

list [illegible] str [illegible]

[illegible] `split()` [illegible] `join()` [illegible]

str.split()

[illegible] str [illegible] list[illegible] str=""[illegible]

[illegible]

```
>>>help(str.split)
```

[illegible]

> split(...) S.split([sep [,maxsplit]]) -> list of strings
>
> Return a list of the words in the string S, using sep as the delimiter string. If maxsplit is given, at most maxsplit splits are done. If sep is not specified or is None, any whitespace string is a separator and empty strings are removed from the result.

[illegible]

```
>>> line = "Hello.I am qiwsir.Welcome you."

>>> line.split(".")      #[illegible] list
['Hello', 'I am qiwsir', 'Welcome you', '']
```

```
>>> line.split(".",1)   #□□ 1,□□□□□□□□□□If maxsplit is given, at
most maxsplit splits are done.
['Hello', 'I am qiwsir.Welcome you.']

>>> name = "Albert Ainstain"    #□□□□□□□□□□□□□
>>> name.split(" ")
['Albert', 'Ainstain']
```

□□□□□□□□□□□□□□□

```
>>> s = "I am, writing\npython\tbook on line"   #□□□□□□□□□□□□□□
□\n□tab □□\t □□
>>> print s         #□□□□□□□
I am, writing
python  book on line
>>> s.split()       #□ split(),□□□□□□□□□□□□
['I', 'am,', 'writing', 'Python', 'book', 'on', 'line']
```

□□ split()□□□□□□□□□□□□□□□□□□□□□□□□□□□□

"[sep]".join(list)

join □□□□ split □□□□□□□□

```
>>> name
['Albert', 'Ainstain']
>>> "".join(name)       #□ list □□□□□□□□□□□□□□□□□□□□□□□□□□
'AlbertAinstain'
>>> ".".join(name)      #□□□□□□□□□□□□□
'Albert.Ainstain'
>>> " ".join(name)      #□□□□□□□□□□□
'Albert Ainstain'
```

□□□□□□□□□□□□□□□□□□□ join.

```
>>> s = "I am, writing\npython\tbook on line"
>>> print s
I am, writing
python  book on line
>>> s.split()
['I', 'am,', 'writing', 'Python', 'book', 'on', 'line']
>>> " ".join(s.split())         #□□□□□□□□□□□□□am □□□□□□□□□□□□□□
'I am, writing python book on line'
```

□□□

□□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

□□

□□

□□□□□□□

```
>>># □□□□ str
>>> s = "abc"
>>> s
'abc'

>>>#□□□□□□□□...
>>> t = 123,'abc',["come","here"]
>>> t
(123, 'abc', ['come', 'here'])
```

□□□□□□□□□□ t□□□□□□□□□□“□□□□□□”□□□□□□□□□□□□□□□□□tuple□□□□□□□□□□□ t□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□ Python □□□

> □□□□□□□□□□□□“□”□□□□□“□”□□□□
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> print "I love %s, and I am a %s" % ('python', 'programmer')
I love Python, and I am a programmer
```

> □□□□□□□□□□□□□

□□□tuple □□□□□□□□□□□□□□ list/str □□□□□□□□□□□□□□□□□□□□□□ list □□□□□□ str □□□□□□□□□□□□□□□□□□□□□□ list □□□□□□□ str□

```
>>> t = 1,"23",[123,"abc"],("python","learn")    #□□□□□□□ list
>>> t
(1, '23', [123, 'abc'], ('python', 'learn'))

>>> t[0] = 8                                    #□□□□□□□□ str
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
```

```
>>> t.append("no")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'tuple' object has no attribute 'append'
    >>>
```

□□□□□□□□□□□□□□□tuple □□□□□□□□□ list □□□ str □□□□□□□□□□□□□

□□□□□

□□□

□□□

```
 >>> t
(1, '23', [123, 'abc'], ('python', 'learn'))
>>> t[2]
[123, 'abc']
>>> t[1:]
('23', [123, 'abc'], ('python', 'learn'))

>>> t[2][0]     #□□□□□□□□□□list □□□□□
123
>>> t[3][1]
'learn'
```

□□□□□□□□□ tuple □□□□□□□□□□□□□□□□□□□

□□

```
 >>> a = (3)
>>> type(a)
<type 'int'>

>>> b = (3,)
>>> type(b)
<type 'tuple'>
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ `type()` □□□□□□□□□□□□□□

□□□ list □□□□□ list □□□□□ tuple □□□□□

□□□ list()□ tuple()□□□□□□□□□:

```
>>> t
(1, '23', [123, 'abc'], ('python', 'learn'))
>>> tls = list(t)                           #tuple-->list
>>> tls
[1, '23', [123, 'abc'], ('python', 'learn')]

>>> t tuple = tuple(tls)                    #list-->tuple
>>> t tuple
(1, '23', [123, 'abc'], ('python', 'learn'))
```

tuple □□□□□

□□□□ list □ str □□□□□□□□□□□□□□□ list □ str □□□□□□

□□□□□□□□□□ list □ str □□,□□□□□□□□,□□□□□□□□□□□□□□□□□□.

□□□□,tuple □□□□□,□□□□□□□□□□:

- Tuple □ list □□ tuple □□ list□
- □□□□□□□□□□□□□ “□□□”□□□□□□□□□□□□ tuple □□□ list □□□□□□□□□ assert □□□□□□□□□□□□□□□□□□□□□□□□□□□□ tuple □ list □□□ (□□□□□□□□□□□□)□
- Tuples □□□ dictionary□□□□□□□□□□ □□□□ key□□□ list □□□Dictionary key □□□□□□□□Tuple □□□□□□□□□□□□□□□□□ list □ tuple□□□□□□□□□□□□□ dictionary key □□□□□□□□□□□□□□□□□□□ dictionary □□□ tuple □□□□□ dictionary key□
- Tuples □□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

[illegible](1)

[illegible] web [illegible] 1.01 [illegible]

> [illegible] 1953 [illegible] 1953 [illegible] 1957 [illegible] 1956 [illegible] 10 [illegible] 200 [illegible]

[illegible]

[illegible]

[illegible] Python [illegible] dictionary [illegible] “[illegible]” [illegible] dict [illegible]

[illegible] 0512 [illegible] 0315 [illegible] 011 [illegible] 012 [illegible]

```
>>> citys = ["suzhou", "tangshan", "beijing", "shanghai"]
>>> city_codes = ["0512", "0315", "011", "012"]
```

[illegible]

```
>>> print "{} : {}".format(citys[0], city_codes[0])
suzhou : 0512
```

> [illegible] city_codes [illegible]

```
>>> suzhou code = 0512
>>> print suzhou_code
330
```

> [illegible] Python [illegible] 0512 [illegible] print [illegible] `int()`, `bin()`, `oct()`, `hex()`

[illegible] Python [illegible] Python [illegible] (dict) [illegible]

□□ dict

□□ 1:

□□□□□□ dict□□□□ dict□□□□□□□□□□□□□□

```
>>> mydict = {}
>>> mydict
{}
```

□□□□“□”□“□□□□□□□□□”□□□□□□“□”□□□□□□□□“□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□”□□□□□□

□□□□□□ dict□

```
>>> person =
{"name":"qiwsir","site":"qiwsir.github.io","language":"python"}
>>> person
{'name': 'qiwsir', 'language': 'python', 'site':
'qiwsir.github.io'}
```

`"name":"qiwsir"`□□□□□□□□□□□□□□□□□ name □□□□key□□□□□ qiwsir □□□□□□□□□□(value)□□□□ dict □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□(:)□□□□□□□□□□□□□□□□□□□(,)□□□

```
>>> person['name2']="qiwsir"    #□□□□□ dict □□□□□□□□□
>>> person
{'name2': 'qiwsir', 'name': 'qiwsir', 'language': 'Python',
'site': 'qiwsir.github.io'}
```

□□□□□□□□□□□ dict □□□□□□□□“□□□”□□□

```
>>> ad = {}
>>> id(ad)
3072770636L
>>> ad["name"] = "qiwsir"
>>> ad
{'name': 'qiwsir'}
>>> id(ad)
3072770636L
```

□□□□□□□□□□□□□□□□□□□□□

□□ 2:

□□□□□□□□□□□□□□□

```
>>> name = (["first","Google"],["second","Yahoo"])
>>> website = dict(name)
>>> website
{'second': 'Yahoo', 'first': 'Google'}
```

□□□□□□□□□

```
>>> ad = dict(name="qiwsir", age=42)
>>> ad
{'age': 42, 'name': 'qiwsir'}
```

□□ **3:**

□□□□□□□□□□□□□□□ fromkeys

```
>>> website = {}.fromkeys(("third","forth"),"facebook")
>>> website
{'forth': 'facebook', 'third': 'facebook'}
```

□□□□□□□□□□□□□□□□□□ dict□

□□□□□□□□□□□□□□“□”□□□□□□□□□□□□□“□”□□□□□□□□□□

```
>>> dd = {(1,2):1}
>>> dd
{(1, 2): 1}
>>> dd = {[1,2]:1}
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unhashable type: 'list'
```

□□ dict □□

dict □□

> □□□□□□□“□□”□“□□”□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□A □□□□□□A □ B □□□□□□A □□□□□□□□□ B □□□□□□□

□□□□□□□□□□□□□□“□”□□□□□□□“□”□

```
>>> person
{'name2': 'qiwsir', 'name': 'qiwsir', 'language': 'python',
'site': 'qiwsir.github.io'}
>>> person['name']
'qiwsir'
```

```
>>> person['language']
'python'
```

□□□□□□□□□“□”□□□□ dict □□“□”□□□“□”□□□□ dict □□“□”□□□“□”□□□□□ dict □□“□”□

□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> city_code = {"suzhou":"0512", "tangshan":"0315", "beijing":"011", "shanghai":"012"}
>>> print city_code["suzhou"]
0512
```

□□ dict □□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□ city_code

```
>>> city_code
{'suzhou': '0512', 'beijing': '011', 'shanghai': '012', 'tangshan': '0315'}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□ list □□
□□□□

> dict □□Python □□□□□□□□
> □□□□□□□□□□□□□□□□□ dict □□□□“□”□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□
> □□□□□□□□□□□□□□□□□□□□□mongdb □□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- len(d)□□□□□(d)□□□□□□□□
- d[key]□□□□□(d)□□□(key)□□
- d[key]=value□□□(value)□□□□(d)□□□(key)
- del d[key]□□□□□(d)□□(key)□□□□□□□□□□
- key in d□□□□(d)□□□□□□□ key □□

□□□□□□□□□

```
>>> city_code
{'suzhou': '0512', 'beijing': '011', 'shanghai': '012', 'tangshan': '0315'}
>>> len(city_code)
4
```

□ city_code □□□□□□len(city_code)□□□ 4□□□□□□□□□□□□□□□□□□

```
>>> city_code["nanjing"] = "025"
>>> city_code
{'suzhou': '0512', 'beijing': '011', 'shanghai': '012',
'tangshan': '0315', 'nanjing': '025'}
```

□□□□□□□

```
>>> city_code["beijing"] = "010"
>>> city_code
{'suzhou': '0512', 'beijing': '010', 'shanghai': '012',
'tangshan': '0315', 'nanjing': '025'}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> city_code["shanghai"]
'012'
>>> del city_code["shanghai"]
```

□□ `city_code["shanghai"]` □□□□□□□(key)□□□□□(value)□□□□□□□□□□□□□□□
del□□□□□□□□□

```
>>> city_code["shanghai"]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'shanghai'
>>> "shanghai" in city_code
False
```

□□□□"shanghai"□□□□□□□□□□□□□□□□□□□□□ `in` □□□□□□□□ `False`□

```
>>> city_code
{'suzhou': '0512', 'beijing': '010', 'tangshan': '0315',
'nanjing': '025'}
```

□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□(4)□□
(4)□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> city_code = {"suzhou":"0512", "tangshan":"0315",
"hangzhou":"0571"}
>>> " Suzhou is a beautiful city, its area code is %(suzhou)s" %
```

```
city code
' Suzhou is a beautiful city, its area code is 0512'
```

[illegible]

[illegible]——[illegible]

[illegible] HTML [illegible] Python [illegible]

```
>>> temp = "<html><head><title>%(lang)s<title><body><p>My name
is %(name)s.</p></body></head></html>"
>>> my = {"name":"qiwsir", "lang":"python"}
>>> temp % my
'<html><head><title>python<title><body><p>My name is qiwsir.</p>
</body></head></html>'
```

temp [illegible] HTML [illegible] dict [illegible]

[illegible]

> [illegible] HTML? [illegible]
>
> [illegible]HyperText Markup Language[illegible]HTML[illegible]HTML [illegible]——[illegible] 1982 [illegible]·[illegible]-[illegible] IETF [illegible] SGML[illegible] HTML[illegible]W3C[illegible]
>
> HTML [illegible] HTML5 [illegible] HTML [illegible]“[illegible]”[illegible] PC[illegible] PAD [illegible]“[illegible]”[illegible] PC [illegible]

[illegible]

[illegible]

> [illegible]Associative Array[illegible]Map[illegible]Dictionary[illegible] C++ [illegible] multimap[illegible] C++ [illegible] map[illegible]
>
> [illegible]
>
> > 1. [illegible]
> > 2. [illegible]

> > 3. [illegible]
> >
> > 4. [illegible]
>
> [illegible]
>
> [illegible] Content-addressable memory [illegible]

[illegible]

> [illegible]Hash table[illegible]Key value[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible](2)

[illegible]

[illegible]

copy

[illegible] copy [illegible]“[illegible]”[illegible] DOS [illegible]“[illegible]”[illegible]“copy”[illegible]“kao([illegible])bei([illegible])”[illegible]“[illegible]”[illegible]

[illegible]copy [illegible] Python [illegible]copy [illegible]

```
>>> a = 5
>>> b = a
>>> b
5
```

[illegible] 5 [illegible] `id()` [illegible]

```
>>> id(a)
139774080
>>> id(b)
139774080
```

[illegible] 5[illegible] Python [illegible] dict [illegible]

```
>>> ad = {"name":"qiwsir", "lang":"Python"}
>>> bd = ad
>>> bd
{'lang': 'Python', 'name': 'qiwsir'}
>>> id(ad)
3072239652L
>>> id(bd)
3072239652L
```

[illegible]“[illegible]”[illegible] copy [illegible]

```
>>> cd = ad.copy()
>>> cd
{'lang': 'Python', 'name': 'qiwsir'}
>>> id(cd)
3072239788L
```

[illegible] cd [illegible] ad [illegible] cd[illegible] ad [illegible]

```
>>> cd["name"] = "itdiffer.com"
>>> cd
{'lang': 'Python', 'name': 'itdiffer.com'}
>>> ad
{'lang': 'Python', 'name': 'qiwsir'}
```

[illegible]“[illegible]”[illegible] Python [illegible]

```
>>> bd
{'lang': 'Python', 'name': 'qiwsir'}
>>> bd["name"] = "laoqi"
>>> ad
{'lang': 'Python', 'name': 'laoqi'}
>>> bd
{'lang': 'Python', 'name': 'laoqi'}
```

[illegible] bd [illegible]“[illegible]”[illegible] ad [illegible]“[illegible]”[illegible]

[illegible]

```
>>> x = {"name":"qiwsir", "lang":["Python", "java", "c"]}
>>> y = x.copy()
>>> y
{'lang': ['Python', 'java', 'c'], 'name': 'qiwsir'}
>>> id(x)
3072241012L
>>> id(y)
3072241284L
```

y [illegible] x [illegible]

```
>>> y["lang"].remove("c")
```

[illegible] y [illegible] dict [illegible]"lang"[illegible] ['Python', 'java', 'c'][illegible] `remove()` [illegible]"c"[illegible] ['Python', 'java']

```
>>> y
{'lang': ['Python', 'java'], 'name': 'qiwsir'}
```

[illegible]x[illegible]

```
>>> x
{'lang': ['Python', 'java'], 'name': 'qiwsir'}
```

[illegible]

[illegible]

```
>>> y["name"] = "laoqi"
>>> y
{'lang': ['python', 'java'], 'name': 'laoqi'}
>>> x
{'lang': ['python', 'java'], 'name': 'qiwsir'}
```

[illegible]

[illegible] `id()`

```
>>> id(x)
3072241012L
>>> id(y)
3072241284L
```

x,y [illegible]

```
>>> id(x["lang"])
3072243276L
>>> id(y["lang"])
3072243276L
```

[illegible]

[illegible]

```
>>> id(x["name"])
3072245184L
>>> id(y["name"])
3072245408L
```

[illegible]

[illegible] Python [illegible]Python [illegible] int,str[illegible]Python [illegible]Python [illegible]

□□□

□□□□□□□□□□□□□□□□□□□□□□□□□“□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ Python □□□□□“□□□”(deep copy)□□□□□□□□ `import` □□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> import copy
>>> z = copy.deepcopy(x)
>>> z
{'lang': ['python', 'java'], 'name': 'qiwsir'}
```

□ `copy.deepcopy()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□(deepcopy)□□□□□□□□□□ id() □□□□□□

```
>>> id(x["lang"])
3072243276L
>>> id(z["lang"])
3072245068L
```

□□□□□□□“□”□□□

```
>>> x
{'lang': ['Python', 'java'], 'name': 'qiwsir'}
>>> x["lang"].remove("java")
>>> x
{'lang': ['Python'], 'name': 'qiwsir'}
>>> z
{'lang': ['Python', 'java'], 'name': 'qiwsir'}
```

□□□□□□□□□□□□□□□

```
>>> x["lang"].append("c++")
>>> x
{'lang': ['Python', 'c++'], 'name': 'qiwsir'}
```

□□□□□□□□□□□□□□

clear

□□□□□□□□ help □□□□□□□□

```
>>> help(dict.clear)
```

```
clear(...)
    D.clear() -> None.  Remove all items from D.
```

□□□□□□□□□□□□□□□□□□

```
>>> a = {"name":"qiwsir"}
>>> a.clear()
>>> a
{}
```

□□□ `clear` □□□□□□□□□□□□□□“□”□□□□□□□□□ `del` □□□□□□□□`del` □□□□□□□□□□□□
□□□□□□□“□”□

```
>>> del a
>>> a
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'a' is not defined
```

□□□□□□

□□□□□□□□□□□□□□□□□□ `a = {}`□□□□□□□□□□□□□□□□ a □□□`{}`□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□ Python □□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□
Python □□□□□

get,setdefault

get □□□□□

```
get(...)
    D.get(k[,d]) -> D[k] if k in D, else d.  d defaults to None.
```

□□□□□□□□“if k in D”□□□□□□□□□...(□□□□)□

```
>>> d
{'lang': 'python'}
>>> d.get("lang")
'python'
```

`dict.get()`□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□
`d['lang']`□□□□□□□□`"Python"`□□□□□□□□□□□□□□□□□

```
>>> print d.get("name")
None

>>> d["name"]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'name'
```

□□□ `dict.get()` □ `dict['key']` □□□□

□□□□□□□□□□□□□□□□□□□□□□ None□□□□□□□□□□□□□□

```
>>> d = {"lang":"Python"}
>>> newd = d.get("name",'qiwsir')
>>> newd
'qiwsir'
>>> d
{'lang': 'Python'}
```

□ `d.get("name",'qiwsir')` □□□□□□□□□□□□"name"□□□□□□□□□□□□□"qiwsir"□□□□□□□□□□□`D[k] if k in D, else d.` □□□□□□□□□□□□□□□□□□□□

□□□□□ get □□□□□□□□□□ `D.setdefault(k)` □□□□□□

```
setdefault(...)
    D.setdefault(k[,d]) -> D.get(k,d), also set D[k]=d if k not in D
```

□□□□□□□ `D.get(k,d)`,□□□□□□□□□□□□□□□□□□□□□□□□□□□k□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> d
{'lang': 'Python'}
>>> d.setdefault("lang")
'Python'
```

□□□□□□"lang"□□□□□□□□□□□□

```
>>> d.setdefault("name","qiwsir")
'qiwsir'
>>> d
{'lang': 'Python', 'name': 'qiwsir'}
```

□□"name"□□□□□□□ `d.setdefault("name","qiwsir")` □□□□"qiwsir"□□□□□□□'name':"qiwsir"` □□□□□□□□□

□□□□□□□

```
>>> d.setdefault("web")
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ print □□□□□□□□□□□□□□□ None.□□□□□□□□□□

```
>>> d
{'lang': 'Python', 'web': None, 'name': 'qiwsir'}
```

□□□□"web"□□□□□ None

items/iteritems, keys/iterkeys, values/itervalues

□□□□□□□□□□□ dict □□
□□

```
>>> help(dict.items)

items(...)
    D.items() -> list of D's (key, value) pairs, as 2-tuples
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `D.items()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> dd = {"name":"qiwsir", "lang":"python",
"web":"www.itdiffer.com"}
>>> dd_kv = dd.items()
>>> dd_kv
[('lang', 'Python'), ('web', 'www.itdiffer.com'), ('name',
'qiwsir')]
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□ `items` □□□□□ `iteritems` □□□□□□□□□□□ iter □ items □□□□□□□□□ items □□□□□□□□□□□□□□□□□□□□□ `D.items()` □□□□□□□□□□□□□□□□ iter □□□□□□□□□□(2)□□□□□□□□□“iterable”□□□□□“□□□□”□□□□ iter □□□□□ iterator □□□□□□□□“□□□”□□□□□"iteritems"□□□□□□

```
iteritems(...)
    D.iteritems() -> an iterator over the (key, value) items of
D
```

□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“D.items()”□□□□□□□□□□□

```
>>> dd
{'lang': 'Python', 'web': 'www.itdiffer.com', 'name': 'qiwsir'}
>>> dd_iter = dd.iteritems()
>>> type(dd_iter)
<type 'dictionary-itemiterator'>
>>> dd_iter
<dictionary-itemiterator object at 0xb72b9a2c>
>>> list(dd_iter)
```

```
[('lang', 'Python'), ('web', 'www.itdiffer.com'), ('name',
'qiwsir')]
```

□□□ dd_iter □□□□□□□'dictionary-itemiterator'□□□□□□□□□□□□□□□□□□□□□□□□□□ `list()` □□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□ key □□ value□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> dd
{'lang': 'Python', 'web': 'www.itdiffer.com', 'name': 'qiwsir'}
>>> dd.keys()
['lang', 'web', 'name']
>>> dd.values()
['Python', 'www.itdiffer.com', 'qiwsir']
```

□□□

pop, popitem

□□□□(3)□□□□□□□□□□□□□□□□ `pop` □ `remove` □□□□□□□□□ `list.remove(x)` □□□□□□□□□□□ `list.pop([i])` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□

```
pop(...)
    D.pop(k[,d]) -> v, remove specified key and return the
corresponding value.
    If key is not found, d is returned if given, otherwise
KeyError is raised
```

`D.pop(k[,d])` □□

```
>>> dd
{'lang': 'Python', 'web': 'www.itdiffer.com', 'name': 'qiwsir'}
>>> dd.pop("name")
'qiwsir'
```

□□□□□□"name"□□□□□□"qiwsir"□□□□□□□□□□“'name':'qiwsir'”□□□□□□□□□□□

```
>>> dd
{'lang': 'Python', 'web': 'www.itdiffer.com'}
```

□□□□□□□pop □□□□□□□□□□□□□□□□□□□□□□ pop □□□□□

```
>>> dd.pop()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: pop expected at least 1 arguments, got 0
```

[illegible]

```
>>> dd.pop("name")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'name'
```

[illegible] `D.popitem()` [illegible] `list.pop()` [illegible]list.pop [illegible] `D.popitem()` [illegible]dict [illegible]

```
popitem(...)
    D.popitem() -> (k, v), remove and return some (key, value) pair as a
    2-tuple; but raise KeyError if D is empty.
```

[illegible]

```
>>> dd
{'lang': 'Python', 'web': 'www.itdiffer.com'}
>>> dd.popitem()
('lang', 'Python')
>>> dd
{'web': 'www.itdiffer.com'}
```

[illegible] tuple

```
>>> dd.popitems()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'dict' object has no attribute 'popitems'
```

[illegible]...[illegible] popitem[illegible] s[illegible] `D.items()` [illegible] s[illegible] `D.popitem()` [illegible]

```
>>> dd.popitem()
('web', 'www.itdiffer.com')
>>> dd
{}
```

[illegible]

```
>>> dd.popitem()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'popitem(): dictionary is empty'
```

[illegible]

update

`update()`,[illegible]

```
update(...)
    D.update([E, ]**F) -> None.  Update D from dict/iterable E
and F.
    If E present and has a .keys() method, does:     for k in E:
D[k] = E[k]
    If E present and lacks .keys() method, does:     for (k, v)
in E: D[k] = v
    In either case, this is followed by: for k in F: D[k] = F[k]
```

[illegible]

[illegible] None,[illegible]

```
>>> d1 = {"lang":"python"}
>>> d2 = {"song":"I dreamed a dream"}
>>> d1.update(d2)
>>> d1
{'lang': 'Python', 'song': 'I dreamed a dream'}
>>> d2
{'song': 'I dreamed a dream'}
```

[illegible] d2 [illegible] d1 [illegible] d1 [illegible] d2 [illegible]d2 [illegible]

[illegible]

```
>>> d2
{'song': 'I dreamed a dream'}
>>> d2.update([("name","qiwsir"), ("web","itdiffer.com")])
>>> d2
{'web': 'itdiffer.com', 'name': 'qiwsir', 'song': 'I dreamed a
dream'}
```

[illegible]

has_key

□□□□□□□□□□□□□□□□□□□□

```
has key(...)
    D.has_key(k) -> True if D has a key k, else False
```

□□□□□□□□ `k in D` □□□

```
>>> d2
{'web': 'itdiffer.com', 'name': 'qiwsir', 'song': 'I dreamed a dream'}
>>> d2.has_key("web")
True
>>> "web" in d2
True
```

□□ dict □□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

□□(1)

□□□□□□□□□□□□□□:int/str/bool/list/dict/tuple

□□□□□□.

□□,Python □□□□□□□□□,□□□□□□□□□□.□□□□□□□□,□□□□□□□□□□□,□□□□□□□,□□□□□□□□□□□□.

□□□□□□□□,□□□□□□□□□□□□□□□.

> □□□□□□□□□□□□□□□“□□□□□□□□□□□□□□□□□□□□□□□□”
>
> □□□□□□□□□“□□”

□□□□□□□□,□□□□□□□□□,□□□□□□□□□□:□□□□□□□□□□□□□□□□□□□□□,□□□□□□.

□□,□□□□□□□□□□□□□□□□,□□□□□□□□,□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□□□,□□□□□□...)

- □□□□□□ dir()□□ help()
- google(□□□ Xdu,□□□□□□□)

□□□□□□□□□□□□□

- □□□□□□□ list/str□□□□□□□□□□
- □□□□□□ list/dict□□□□□□/□□□□□□□□□□
- □□□□□□ str/int□□□□□□□□□□
- □□□□□□□□□ dict□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□ set□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□ set

tuple □□ list □ str □□□(□□□□□□□□□□,□□□□□□□□□□□□),□□ set □□□□□□ list □ dict □□□.

set □□□□ dict □□□:□□□{}□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□;□□,set □□□□□□□□,□□□□ dict □□.

set □□□□ list □□□:□□□□□□□□□□□□.

[illegible],[illegible] set [illegible]:

```
>>> s1 = set("qiwsir")
>>> s1
set(['q', 'i', 's', 'r', 'w'])
```

[illegible] str [illegible],[illegible] set.[illegible]:qiwsir [illegible] i[illegible] s1 [illegible],[illegible] i,[illegible]

```
>>> s2 = set([123,"google","face","book","facebook","book"])
>>> s2
set(['facebook', 123, 'google', 'book', 'face'])
```

[illegible] s2 [illegible]

```
>>> s3 = {"facebook",123}        #[illegible]{}[illegible]
>>> s3
set([123, 'facebook'])
```

[illegible] `set()` [illegible] `{}` [illegible]Python [illegible]

```
>>> s3 = {"facebook",[1,2,'a'],
{"name":"Python","lang":"english"},123}
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unhashable type: 'dict'

>>> s3 = {"facebook",[1,2],123}
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unhashable type: 'list'
```

[illegible],[illegible],[illegible]{}[illegible] list/dict [illegible] set.

[illegible]“unhashable”[illegible]“hashable”[illegible]“[illegible]”[illegible]“[illegible]”[illegible](1)[illegible]“[illegible]”(unhashable)[illegible] list/dict[illegible] unhashable[illegible] str [illegible] hashable[illegible]

[illegible] hashable [illegible]

[illegible]“[illegible]”[illegible] list/dict [illegible]

[illegible]:

```
>>> s1
set(['q', 'i', 's', 'r', 'w'])
```

```
>>> s1[1] = "I"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'set' object does not support item assignment
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> s1
set(['q', 'i', 's', 'r', 'w'])
>>> lst = list(s1)
>>> lst
['q', 'i', 's', 'r', 'w']
>>> lst[1] = "I"
>>> lst
['q', 'I', 's', 'r', 'w']
```

□□□ `list()` □ `set()` □□□□□□□□□□□□□□□□

□□□□□□□ `set()` □□□□□□□□□□□□□□□□□□□ unhashable □□□□

set □□□

□□□□□□□□□□□□□□□□□,□ set □□□□□□□□□□,□□□□□□ set □□□□□.

```
 >>> dir(set)
['  and  ', '  class  ', '  cmp  ', '  contains  ',
'  delattr  ', '  doc  ', '  eq  ', '  format  ', '  ge__',
'  getattribute  ', '  gt  ', '  hash  ', '  iand  ',
'  init  ', '  ior  ', '  isub  ', '  iter  ', '  ixor  ',
'  le  ', '  len  ', '  lt  ', '  ne  ', '  new  ', '  or__',
'  rand  ', '  reduce  ', '  reduce ex  ', '  repr  ',
'  ror  ', '  rsub  ', '  rxor  ', '  setattr  ', '  sizeof _',
'  str  ', '  sub  ', '  subclasshook  ', '  xor  ', 'add',
'clear', 'copy', 'difference', 'difference update', 'discard',
'intersection', 'intersection update', 'isdisjoint', 'issubset',
'issuperset', 'pop', 'remove', 'symmetric difference',
'symmetric_difference_update', 'union', 'update']
```

□□□□□□□,□□□□□□__□□□□□□□□(□□□□□□□□□□□□□□):

> 'add', 'clear', 'copy', 'difference', 'difference_update', 'discard', 'intersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'pop', 'remove', 'symmetric_difference', 'symmetric_difference_update', 'union', 'update'

□□□ help()□□□□□□□□□□□□□□□,□□□□□□□:

add, update

```
>>> help(set.add)

Help on method_descriptor:

add(...)
Add an element to a set.
This has no effect if the element is already present.
```

□□□□□□□□□□□□□□□□□□□□:

```
>>> a set = {}               #□□□□□□□□□□□□□□□□ set
>>> a set.add("qiwsir")     #□□.□□□□□□,□□□□□ dict □□ add.□□□□□□□
set □.
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'dict' object has no attribute 'add'
>>> type(a set)             #type □□□□,□□□□□□□□□□□□ dict
<type 'dict'>
```

□□□□□□,{}□□□□,□ dict □ set □□□.□□,□□□□□□□□□□ dict,□□ set.□□ Python □□□.□□□ set,□□□□□□□□□□□.

```
>>> a set = {'a','i'}       #□□□□ set □□
>>> type(a set)
  <type 'set'>              #□□

>>> a set.add("qiwsir")     #□□□□□□
>>> a set                   #□□□□,□□□□ a_set □□□□□□□□
set(['i', 'a', 'qiwsir'])

>>> b set = set("python")
>>> type(b set)
<type 'set'>
>>> b set
set(['h', 'o', 'n', 'p', 't', 'y'])
>>> b set.add("qiwsir")
>>> b set
set(['h', 'o', 'n', 'p', 't', 'qiwsir', 'y'])

>>> b_set.add([1,2,3])      #□□.list □□□□□□□□□□□□□□□□ hashable □
□□
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unhashable type: 'list'
```

```
>>> b set.add('[1,2,3]')    #□□□□!
>>> b set
set(['[1,2,3]', 'h', 'o', 'n', 'p', 't', 'qiwsir', 'y'])
```

□□□□□□□□□□□□□,□□□□□□□□ set □□□□□□□,□□□ set.update(s2)

```
>>> help(set.update)
update(...)
    Update a set with the union of itself and others.

>>> s1
set(['a', 'b'])
>>> s2
set(['github', 'qiwsir'])
>>> s1.update(s2)       #□ s2 □□□□□□ s1 □.
>>> s1                  #s1 □□□□□□□
set(['a', 'qiwsir', 'b', 'github'])
>>> s2                  #s2 □□□
set(['github', 'qiwsir'])
```

pop, remove, discard, clear

```
>>> help(set.pop)
pop(...)
    Remove and return an arbitrary set element.
    Raises KeyError if the set is empty.

>>> b set
set(['[1,2,3]', 'h', 'o', 'n', 'p', 't', 'qiwsir', 'y'])
>>> b set.pop()     #□ set □□□□□□□□,□□□□□
'[1,2,3]'
>>> b_set.pop()
'h'
>>> b_set.pop()
'o'
>>> b set
set(['n', 'p', 't', 'qiwsir', 'y'])

>>> b set.pop("n")  #□□□□□□□□□□□,□□□.
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: pop() takes no arguments (1 given)
```

set.pop()□□ set □□□□□□□□,□□□□□□□□□.□□,□□□□□□□□□□.□□□□□□□□□□□,pop()□□□□□.□□,□□ set □□□□,□□□.□□□□□□□□□□□□□,□□□□□□.

□□□□□□□□,□□□?

```
>>> help(set.remove)

remove(...)
    Remove an element from a set; it must be a member.

    If the element is not a member, raise a KeyError.
```

`set.remove(obj)` □□ obj,□□□ set □□□□,□□□□□.□□□:

```
>>> a_set
set(['i', 'a', 'qiwsir'])
>>> a_set.remove("i")
>>> a_set
set(['a', 'qiwsir'])
>>> a_set.remove("w")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'w'
```

□ remove(obj)□□□□□□□ discard(obj):

```
>>> help(set.discard)

discard(...)
    Remove an element from a set if it is a member.

    If the element is not a member, do nothing.
```

□ `help(set.remove)` □□□□□,□□□□□□□.discard(obj)□□ obj □□□ set □□□□,□□□,□□□□,□□□□□□,do nothing.□□□□□□□□□□□□□.□□□□□.

```
>>> a_set.discard('a')
>>> a_set
set(['qiwsir'])
>>> a_set.discard('b')
>>>
```

□□□□□□□□□□,□□ set.clear(),□□□□□:Remove all elements from this set.(□□□□□□□□□□ help(set.clear))

```
>>> a_set
set(['qiwsir'])
>>> a_set.clear()
>>> a_set
set([])
```

```
>>> bool(a_set)      #[illegible],bool [illegible] False.
False
```

[illegible]

[illegible],[illegible]([illegible])

> [illegible]—[illegible]—[illegible]“[illegible]”[illegible]“[illegible]”[illegible] x [illegible] A [illegible] x ∈ A[illegible]
>
> [illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible],[illegible]?[illegible],[illegible]:

> [illegible] 0 [illegible]
>
> [illegible]

[illegible],[illegible].

[illegible],[illegible].[illegible]/[illegible],[illegible].

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible](2)

[illegible]

[illegible](1)[illegible] `set()` [illegible] unhashable

[illegible] `frozenset()` [illegible] hashable [illegible]——[illegible]

```
>>> f set = frozenset("qiwsir")
>>> f set
frozenset(['q', 'i', 's', 'r', 'w'])
>>> f set.add("python")              #[illegible]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'frozenset' object has no attribute 'add'

>>> a set = set("github")            #[illegible] set
>>> a set
set(['b', 'g', 'i', 'h', 'u', 't'])
>>> a set.add("python")
>>> a set
set(['b', 'g', 'i', 'h', 'python', 'u', 't'])
```

[illegible]

[illegible]

[illegible]

[illegible]

```
>>> aset
set(['h', 'o', 'n', 'p', 't', 'y'])
>>> "a" in aset
False
```

```
>>> "h" in aset
True
```

□□□□□□□□

□□□□□□ A□B

- A □□□□ B□□□□□□□□□□□□□

□□□□□□□□

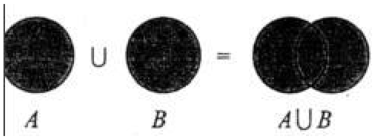
```
 >>> a
set(['q', 'i', 's', 'r', 'w'])
>>> b
set(['a', 'q', 'i', 'l', 'o'])
>>> a == b
False
>>> a != b
True
```

- A □□□ B □□□□□□□□□□B □□□ A □□□□□ A □□□□□□ B □□□□□□ B □□□□ A □□□□□□□

□□□□ A □□□□□ B □□□□□□□□ `A<B`□□□ true □□□□□□□□□□□□□□□□□□□□ `A.issubset(B)`□□□

```
 >>> a
set(['q', 'i', 's', 'r', 'w'])
>>> c
set(['q', 'i'])
>>> c<a     #c □ a □□□
True
>>> c.issubset(a)   #□□□□□□□□□□ c □□□ a □□□
True
>>> a.issuperset(c) #□□ a □□□ c □□□
True

>>> b
set(['a', 'q', 'i', 'l', 'o'])
>>> a<b     #a □□ b □□□
False
>>> a.issubset(b)   #□□□□□
False
```

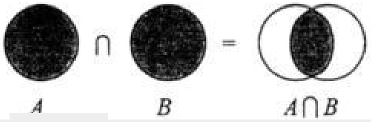
- A□B □□□□□ A□B □□□□□□□□□□



□□□□□□□□□“|”□□□□□□□□□□□□□□□□□□□□□□□□□□□"shift"□□□□□□□□□□□□□□□□□□□□□ `A | B`.□□□□□□ `A.union(B)` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ A □□□

```
>>> a
set(['q', 'i', 's', 'r', 'w'])
>>> b
set(['a', 'q', 'i', 'l', 'o'])
>>> a | b                      #□□□□□□□□□□□□
set(['a', 'i', 'l', 'o', 'q', 's', 'r', 'w'])
>>> a.union(b)
set(['a', 'i', 'l', 'o', 'q', 's', 'r', 'w'])
```

- A□B □□□□□ A□B □□□□□□□□□□□□□

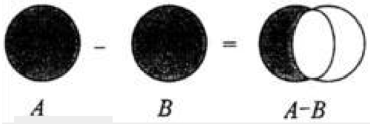


```
>>> a
set(['q', 'i', 's', 'r', 'w'])
>>> b
set(['a', 'q', 'i', 'l', 'o'])
>>> a & b       #□□□□□□□
set(['q', 'i'])
>>> a.intersection(b)
set(['q', 'i'])
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> a and b
set(['a', 'q', 'i', 'l', 'o'])
```

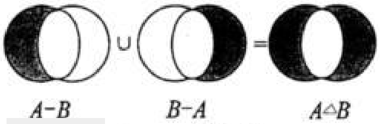
- A □□ B □□□□□□□ A □□ B □□□□□□□□□□□□□□



```
>>> a
set(['q', 'i', 's', 'r', 'w'])
>>> b
set(['a', 'q', 'i', 'l', 'o'])
>>> a - b
```

```
set(['s', 'r', 'w'])
>>> a.difference(b)
set(['s', 'r', 'w'])
```

-A[illegible]B [illegible]

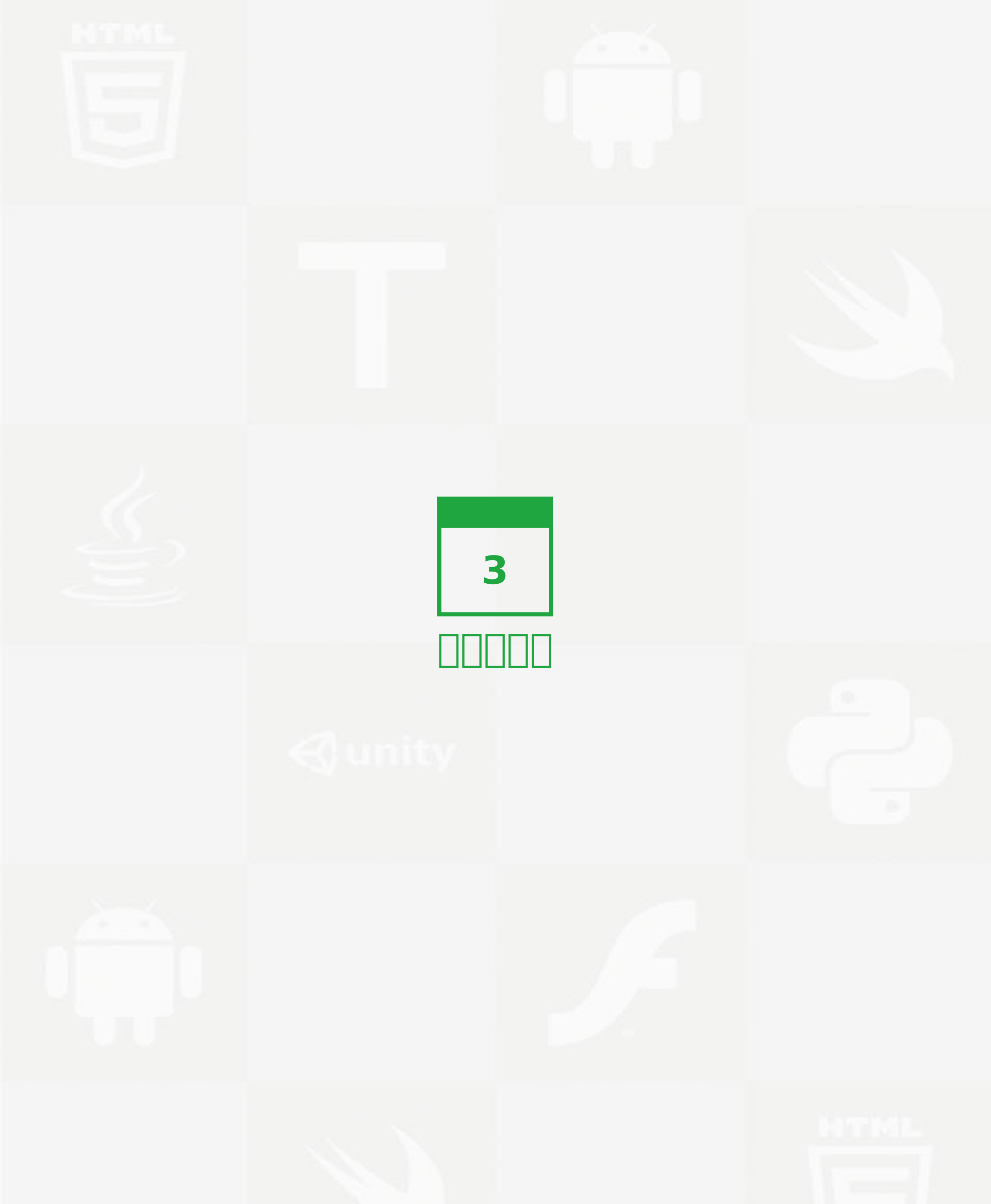


```
 >>> a
set(['q', 'i', 's', 'r', 'w'])
>>> b
set(['a', 'q', 'i', 'l', 'o'])
>>> a.symmetric_difference(b)
set(['a', 'l', 'o', 's', 'r', 'w'])
```

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]



3

[illegible]

[illegible]

[illegible] Python [illegible]

[illegible]

[illegible]

[illegible] + - * / [illegible] % [illegible]

[illegible]

[illegible]	[illegible]	[illegible]
+	[illegible] - [illegible]	10+20 [illegible] 30
-	[illegible] - [illegible]	10-20 [illegible] -10
*	[illegible] - [illegible]	10 * 20 [illegible] 200
/	[illegible] - x [illegible] y	20/10 [illegible] 2
%	[illegible] - [illegible]	20%10 [illegible] 0
**	[illegible] - [illegible] x [illegible] y [illegible]	10**2 [illegible] 100
//	[illegible] - [illegible]	9//2 [illegible] 4 , 9.0//2.0 [illegible] 4.0

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]Python [illegible]

[illegible] a [illegible] 10[illegible] b [illegible] 20[illegible]

[illegible]	[illegible]	[illegible]
==	[illegible] - [illegible]	(a == b) [illegible] False[illegible]
!=	[illegible] - [illegible]	(a != b) [illegible] true.
>	[illegible] - [illegible]x[illegible]y	(a > b) [illegible] False[illegible]
<	[illegible] - [illegible]x[illegible]y	(a < b) [illegible] true[illegible]
>=	[illegible] - [illegible]x[illegible]y[illegible]	(a >= b) [illegible] False[illegible]
<=	[illegible] - [illegible]x[illegible]y[illegible]	(a <= b) [illegible] true[illegible]

[illegible] true [illegible] false[illegible] True[illegible] False[illegible]

[illegible]

```
>>> a=10
>>> b=20
>>> a>b
False
>>> a<b
True
>>> a==b
False
>>> a!=b
True
>>> a>=b
False
```

```
>>> a<=b
True
```

[illegible]“[illegible]”[illegible]

```
>>> a = "qiwsir"
>>> b = "Python"
>>> a > b
True
```

[illegible]q [illegible] p[illegible]q [illegible] p [illegible] True.

[illegible] Python [illegible]

```
>>> a = 5
>>> b = "5"
>>> a > b
False
```

[illegible]

[illegible]

> [illegible]————[illegible]

[illegible] Python [illegible]

> [illegible]logic[illegible]

[illegible][illegible]

[illegible]

[illegible]

> [illegible]·[illegible]George Boole[illegible]1815 [illegible] 11 [illegible]1864 [illegible])[illegible]
>
> [illegible]·[illegible] 19 [illegible]

> [illegible]
>
> 1848 [illegible]The Mathematical Analysis of Logic[illegible]
>
> 1849 [illegible] UCC[illegible]1854 [illegible]The Laws of Thought[illegible] 19 [illegible]
>
> [illegible]

[illegible]

[illegible]“[illegible]”[illegible]True [illegible] False[illegible] 1 [illegible] 0[illegible]

[illegible] 1(True)[illegible]0(False)[illegible]

[illegible] Python [illegible]

[illegible]

[illegible]

[illegible] a [illegible] 10[illegible] b [illegible] 20[illegible]

[illegible]	[illegible]	[illegible]
and	[illegible]"[illegible]" - [illegible] x [illegible] False[illegible]x and y [illegible] False[illegible] y [illegible]	(a and b) [illegible] true[illegible]
or	[illegible]"[illegible]" - [illegible] x [illegible] True[illegible] True[illegible] y [illegible]	(a or b) [illegible] true[illegible]
not	[illegible]"[illegible]" - [illegible] x [illegible] True[illegible] False[illegible] x [illegible] False[illegible] True[illegible]	not(a and b) [illegible] false[illegible]

- and

and[illegible]“[illegible]”[illegible]A and B[illegible]A[illegible] A [illegible] true[illegible] B[illegible] B [illegible] B [illegible] False[illegible] A and B [illegible] False,[illegible] B [illegible] True[illegible] A and B [illegible] True[illegible] A [illegible] False ,[illegible] B [illegible] A and B [illegible] False.

□□□

`4>3 and 4<9`□□□□ `4>3` □□□□□□□ `True`□□□ `4<9` □□□□ `True`□□□□□□□□□□□ □□□ `True`.

```
>>> 4>3 and 4<9
True
```

`4>3 and 4<2`□□□ `4>3`□□□ `True`□□□ `4<2`□□□□□ `False`□□□□□□□□ `False`.

```
>>> 4>3 and 4<2
False
```

`4<3 and 4<9`□□□ `4<3`□□□□ `False`,□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> 4<3 and 4<2
False
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ and □□□□□□□ true □□ true□□□□□ false □□□ false□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- or

or□□□□“□”□□□□ A or B □□□□□□□□□□

```
if A==True:
    return True
else:
    if B==True:
        return True
    else if B==False:
        return False
```

□□□

□□

```
if A==True:         #□□ A □□□ True
    return True     #□□ True,□□□□□□□□ True
else:               #□□□□□□ A □□□□ True
    if B==True:     #□ B □□□□□□□□B□□□□□□□□□
        return True
    else if B==False:
        return False
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> 4<3 or 4<9
True
>>> 4<3 or 4>9
False
>>> 4>3 or 4>9
True
```

- not

not□□□□“□”□□□□□□□□□□□□□□□□□□□□□□

```
>>> not(4>3)
False
>>> not(4<3)
True
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ in□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

[illegible](1)

[illegible]

[illegible]“[illegible]”[illegible]

[illegible]

[illegible]`print "Hello, World"`[illegible]

[illegible]

> [illegible]Imperative programming[illegible]
>
> [illegible]

[illegible]Python [illegible]

- [illegible]:[illegible]
- [illegible]:[illegible]
- [illegible] Goto[illegible] Procedure [illegible]

[illegible]

[illegible]Python [illegible] Python [illegible]

print

□ Python2.x □□print □□□□□□□□□□ Python3.x □□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□ Python2.x□

> □□□□□ Python3.x□□□
> Python2.x□Python3.x □□ Python2.x □□□□□□□□□□□□ Python □□□□□□□□□□□□□□
> □□□□□□□□□□□□□□ Python2.x□□□□□□□□ Python2.x□□□□□□□ Python3.x□□□□□□□□
> □□□□□□□

print □□□
□ print □□

```
>>> print "hello, world"
hello, world
>>> print "hello","world"
hello world
```

□□□□□□□□□□□ print □□□□□□□□□□□□□□□"hello, world"□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□"hello"□□□□□□□□"world"□□□□□□□□□□□□□□□□

□□□□ print □□□□□□□□□□□□□□□ `\n` □□
□□□□□□□□□"hello"□"world"□□□□□□□□

□□□

```
>>> for i in [1,2,3,4,5]:
...     print i
...
1
2
3
4
5
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ i□□□ print □□□□□□□□□□□□□ i □□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□

```
>>> for i in [1,2,3,4,5]:
...     print i ,
...
1 2 3 4 5
```

□□□ print □□□□□□□□□□□□□□□□□□□□□□□□□□□

print □□

import

[illegible] math [illegible] Python [illegible] math [illegible] import [illegible]

[illegible] import [illegible] Python [illegible]

```
>>> import math
>>> math.pow(3,2)
9.0
```

[illegible] `math.pow(3,2)` [illegible]pow()[illegible] math [illegible]

```
>>> from math import pow
>>> pow(3,2)
9.0
```

[illegible] pow()[illegible] math [illegible] `pow()` [illegible]

```
>>> from math import pow as pingfang
>>> pingfang(3,2)
9.0
```

[illegible] pow [illegible] pingfang[illegible] `pingfang()` [illegible] `pow()` [illegible]

[illegible]

```
>>> from math import pow, e, pi
>>> pow(e,pi)
23.140692632779263
```

[illegible] math [illegible] pow,e,pi[illegible]pow()[illegible]e[illegible]pi [illegible] π.

> e[illegible]Euler's number[illegible]·[illegible]e = 2.71828182845904523536([illegible])
>
> e [illegible] π [illegible],[illegible] e [illegible] π [illegible]([illegible])

```
>>> from math import *
>>> pow(3,2)
9.0
>>> sqrt(9)
3.0
```

[illegible] math [illegible]

[illegible] math [illegible]

[illegible]

[illegible] `a = 3` [illegible]

> [illegible]“=”[illegible]“=”[illegible]“=”[illegible]

[illegible]

```
>>> x, y, z = 1, "python", ["hello", "world"]
>>> x
1
>>> y
'python'
>>> z
['hello', 'world']
```

[illegible]

```
>>> a = "itdiffer.com", "python"

>>> a
('itdiffer.com', 'python')
```

[illegible] a[illegible] Python [illegible]

[illegible] Python [illegible]

[illegible] `a = 2`,`b = 9`[illegible] `a = 9`,`b = 2`.

[illegible]

```
temp = a;
a = b;
b = temp;
```

[illegible] a [illegible] 2[illegible](temp)[illegible] a [illegible] b [illegible]([illegible] 9)[illegible] a [illegible](a = b)[illegible]b [illegible] 2 [illegible] b [illegible]

[illegible]

Python [illegible]

```
>>> a = 2
>>> b = 9

>>> a, b = b, a

>>> a
9
>>> b
2
```

`a, b = b, a` [illegible] Python [illegible]

[illegible]“[illegible]”

```
>>> m = n = "I use python"
>>> print m,n
I use python I use python
```

[illegible]

```
>>> id(m)
3072659528L
>>> id(n)
3072659528L
```

[illegible] `id()` [illegible]

[illegible] `id()` [illegible]

```
>>> m is n
True
```

□□□□□ m □ n □□□□□□□□□□□□□□True □□□□□□□

```
>>> a = "I use python"
>>> b = a
>>> a is b
True
```

□□□□□□□□□□□□□

□□□

```
>>> b = "I use python"
>>> a is b
False
>>> id(a)
3072659608L
>>> id(b)
3072659568L

>>> a == b
True
```

□□□□□□□□□□□□ a□b □□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□`x = x + 1`□□□□□□□□□□□□□□□□□□□□□□“=”□□□□□□□□□□□□□□□□□□□□□□"="□□□□□□□□□□□ x □□ 1 □□□□□□□□□□□□□□x.

□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□”□+□-□*□/□% □□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`x += 1`

```
>>> x = 9
>>> x += 1
>>> x
10
```

□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> m = "py"
>>> m += "th"
>>> m
'pyth'
>>> m += "on"
>>> m
'python'
```

□□□

□□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

[illegible](2)

[illegible]

if

if[illegible]conj. [illegible]if [illegible]

```
>>> a = 8
>>> if a==8:
...     print a
...
8
```

[illegible]if[illegible]

`if a==8:`[illegible] `a==8` [illegible] True[illegible] `print a` [illegible] Python [illegible]

[illegible]

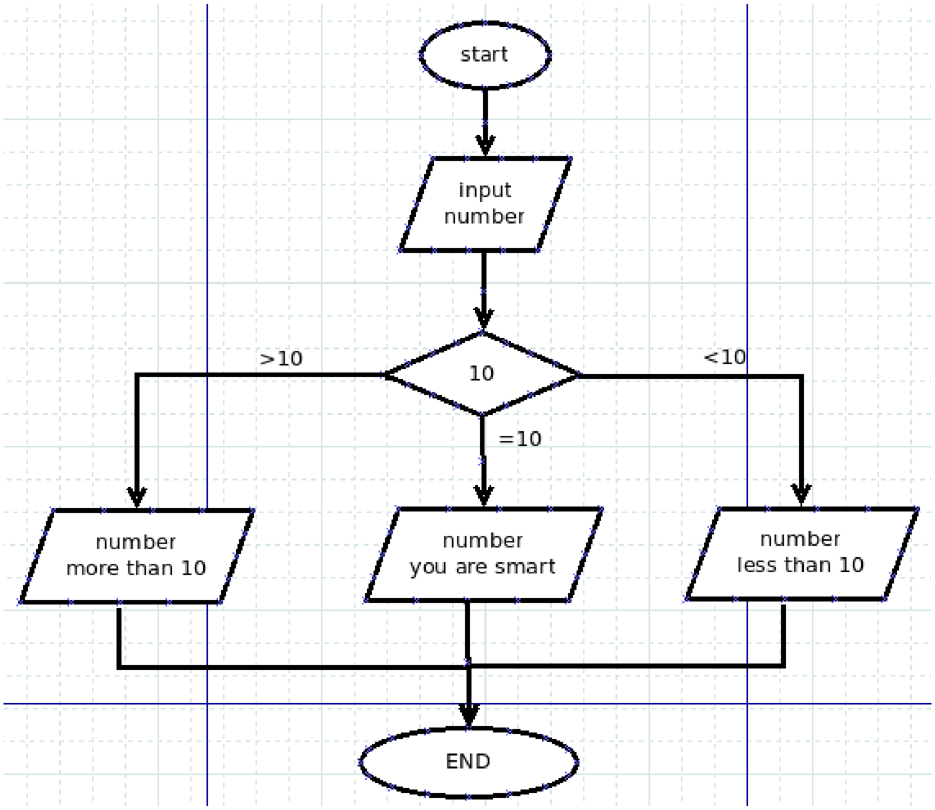
> Python [illegible]Python [illegible]Off-side [illegible] if [illegible].
>
> [illegible] PEP [illegible] 4 [illegible] Tab [illegible] Tab [illegible] Python [illegible]

[illegible]

- [illegible]
- [illegible]

if/else/elif

□□□□□□□□□□□□□ if□□□□□□□□□□□□□□□□□


start
input number
10
>10
<10
=10
number more than 10
number you are smart
number less than 10
END

□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ 10□□□□□□□□□ 10,□□□ 10□□□□□□□□□ 10,□□□□ 10,□□□□□□□□□□

□□□□□□□□□□□□□□□ if □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ if...elif...else □□□

□□□□□□□

```
 if □□ 1:
    □□□□□ 1
elif □□ 2:
    □□□□□ 2
elif □□ 3□
    □□□□□ 3
else:
    □□□□□ 4
```

elif □□□□□□□□□□□□□□□□□□□□□□□ if□□□□ else□

[illegible][illegible][illegible]

[illegible]

```
#! /usr/bin/env python
#coding:utf-8

print "[illegible]"

number = int(raw_input())   #[illegible] raw_input()[illegible]
                            #[illegible] int()[illegible]

if number == 10:
    print "[illegible]%d"%number
    print "You are SMART."
elif number > 10:
    print "[illegible]%d"%number
    print "This number is more than 10."
elif number < 10:
    print "[illegible]%d"%number
    print "This number is less than 10."
else:
    print "Are you a human?"
```

[illegible] raw_input() [illegible]

[illegible]number == 10[illegible] number [illegible] == [illegible] 10,[illegible] int [illegible]number [illegible] int [illegible].

[illegible].py [illegible] `num.py`,[illegible] `Python num.py` [illegible]

```
$ Python num.py
[illegible]
12
[illegible]12
This number is more than 10.

$ Python num.py
[illegible]
10
[illegible]10
```

```
You are SMART.

$ Python num.py
[illegible]
9
[illegible]9
This number is less than 10.
```

[illegible]

```
#! /usr/bin/env python
```

[illegible]

[illegible] # [illegible] Python [illegible] /usr/bin Python[illegible] Python [illegible] /usr/bin [illegible] /usr/bin/env[illegible] Python [illegible] Unix [illegible] windows [illegible]

[illegible]“[illegible]”[illegible] True[illegible]

[illegible]

[illegible]

```
>>> name = "qiwsir" if "laoqi" else "github"
>>> name
'qiwsir'
>>> name = 'qiwsir' if "" else "python"
>>> name
'Python'
>>> name = "qiwsir" if "github" else ""
>>> name
'qiwsir'
```

[illegible]A = Y if X else Z

[illegible]

- [illegible] X [illegible] A=Y
- [illegible] X [illegible] A=Z

[illegible]

```
>>> x = 2
>>> y = 8
>>> a = "python" if x>y else "qiwsir"
>>> a
'qiwsir'
>>> b = "python" if x<y else "qiwsir"
>>> b
'python'
```

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□(3)

□□

□□□□□□□□□□□□□□□□□□□□□□□□□

□ Python □□□□□□□□□□for □□□

□□□□□□□

```
for □□□□□
    □□□□
```

□□□□□□□□□□□□ if □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ for □□□□

□□□□ print □□□□□□□□□□□□□□□□□□□□□□□□

```
>>> hello = "world"
>>> for i in hello:
...     print i
...
w
o
r
l
d
```

□□ for □□□□□□□□□□

1. hello □□□□□□□□"world"□□ str □□□□□
2. □□ i □□ hello □□□□□□□□□"world",□□ str □□
3. □ i="w"□□□□□□ print i□□□□□□□ w□□□□□□□□□□□□ i="e"□□□□□ print i,□□□□□ e□□□

[illegible][illegible](1)[illegible][illegible] print [illegible]

[illegible]

```
>>> for i in range(len(hello)):
...     print hello[i]
...
w
o
r
l
d
```

[illegible]

1. len(hello)[illegible] hello [illegible] 5
2. range(len(hello),[illegible] range(5),[illegible][0, 1, 2, 3, 4],[illegible]"world"[illegible] `range()` [illegible]
3. for i in range(len(hello)),[illegible] for i in [0,1,2,3,4],[illegible]i[illegible] list [illegible] i=0 [illegible] hello[0][illegible] i=4 [illegible]

[illegible] str [illegible] list[illegible] list [illegible]

```
>>> ls_line
['Hello', 'I am qiwsir', 'Welcome you', '']
>>> for word in ls_line:
...     print word
...
Hello
I am qiwsir
Welcome you

>>> for i in range(len(ls_line)):
...     print ls_line[i]
...
Hello
I am qiwsir
Welcome you
```

range(start,stop[, step])

[illegible]`range(start, stop[, step])`

□□□ Python □□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□

> range
>
> n. □□□□□□□□□□ vi. □□...□□□□□□□□□□□□□□□□□□□□□□□□ vt. □□□□□□□□□□□□□□
> □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

> This is a versatile function to create lists containing arithmetic progressions. It is most often used in for loops. The arguments must be plain integers. If the step argument is omitted, it defaults to 1. If the start argument is omitted, it defaults to 0. The full form returns a list of plain integers [start, start + step, start + 2 * step, ...]. If step is positive, the last element is the largest start + i * step less than stop; if step is negative, the last element is the smallest start + i * step greater than stop. step must not be zero (or else ValueError is raised).

□□□□□□□□□□□□□ `range()` □□□□□□□□

- □□□□□□□□□□□□□□□□□□□□□
- □□□□□□□□ for □□□□□□□ for □□□□□□□□□□□□
- □□□□□□□□□□□□□□□ 0 □□□□□□□□□[start, start + step, start + 2*step, ...]□□□□
- step □□□□ 1□□□□□□□□□□□□□
- □□ step □□□□□□ list □□□□□□□□□ stop □□□ start+i*step □□□□□ stop□□□ step □□□□start+i*step □□□□ stop□
- step □□□□□□□□□□□□□□□□

□□□□□□□□□□□ range(start,stop[,step])□□□□

- start□□□□□□□□□ 0,□□□□□□□□□□□□□□ start=0
- stop□□□□□□□□□□□□□
- step□□□□□□□□□□ 1,□□□□□□□□□□□□□ 1□□□□□□ 0

□□□□□□□□□□□□□□□□□

```
>>> range(9)                  #stop=9□□□□□□□□□□□□
range(0,9,1)
[0, 1, 2, 3, 4, 5, 6, 7, 8] #□ 0 □□□□□□ 1,□□□□□□□ 9 □□□□
>>> range(0,9)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8]
>>> range(0,9,1)
[0, 1, 2, 3, 4, 5, 6, 7, 8]

>>> range(1,9)              #start=1
[1, 2, 3, 4, 5, 6, 7, 8]

>>> range(0,9,2)            #step=2,□□□□□□ start+i*step□
[0, 2, 4, 6, 8]
```

□□□□□□ range(0,9,2)

- □□□□ 0 □□□□□□ 1,□□□□ range(9)□□□□□□□□□□□□ 2□□□ range(9,2)□□□□□□□□□□□□□□□□□□ start=9,stop=2□□□□□□□□□ 1 □□□□□□□□□ start □□□□□□
- start=0,step=2,stop=9.list □□□□□□□ start=0,□□□□□ start+1*step=2 □□□□□□□ 1□□□ 2□□□□□□□□□□□□□□□□ list □□ str□□□□□□□□□□□□□□□ 0 □□□□□□ n □□□□ start+(n-1)*step□□□□□ stop □□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> range(-9)
```

□□□□□□□□□[0,-1,-2,-3,-4,-5,-6,-7,-8],□□□□□□□□□

□□□□□□□ start=0,step=1,stop=-9.

□□□□□ 0□□□□□ start+1*step□□□□□□□□□□□□ 1,□□□□□□□□ -9□□□□□□□□□□□□Python □□□□□□□□□□□□□□□□

```
>>> range(-9)
[]
>>> range(0,-9)
[]
>>> range(0)
[]
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> range(0,-9,-1)
[0, -1, -2, -3, -4, -5, -6, -7, -8]
>>> range(0,-9,-2)
[0, -2, -4, -6, -8]
```

□□□□□□□□□□□□□□□□□□□□□□

```
>>> range(0,100,2)
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30,
32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60,
62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90,
92, 94, 96, 98]
```

100 □□□□□□□□□□□□ list□□□□□□□□□□

□□□□□□□□□□□□□□□□ ["I","am","a","Pythoner","I","am","learning","it","with","qiwsir"],□□□□ list □□□□□□□ list□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

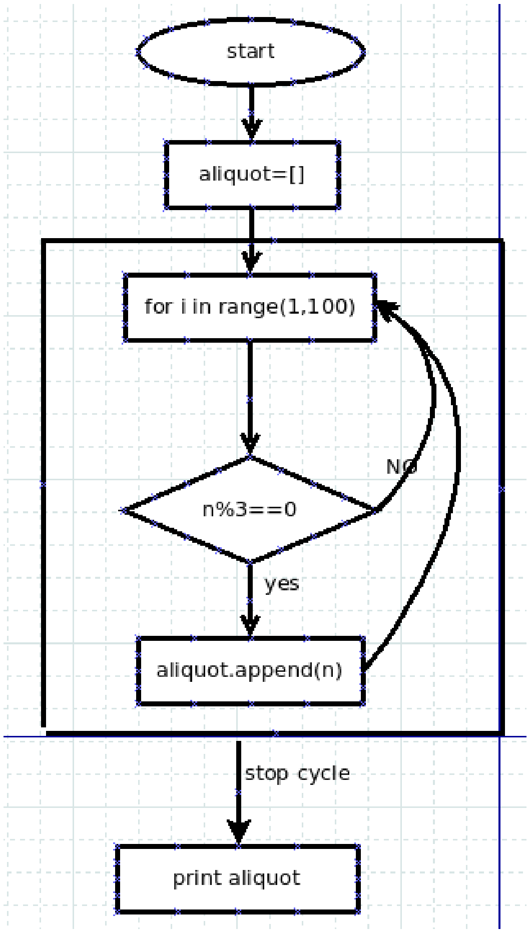
```
>>> pythoner
['I', 'am', 'a', 'pythoner', 'I', 'am', 'learning', 'it',
'with', 'qiwsir']
>>> py_index = range(len(pythoner))     #□ len(pythoner)□
stop □□
>>> py_index
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

□□□□□□ Pythoner □□□□□□□□□□□□□□□□□

□□□□ 100 □□□□□ 3 □□□□□□

□□□□□□□□□□□□□□□□□ 100 □□□□□□□□□□□□□□□ range(1,100)□□□□□□□□□ 3 □□□□□□□□□□□□ n□□□□□□□□ 3 □□□□□ n%3(% □□□□)□ 0.□□□□□□ n □□□□□□ for □□□

□□


start
aliquot=[]
for i in range(1,100)
n%3==0
NO
yes
aliquot.append(n)
stop cycle
print aliquot

[illegible]

[illegible]

```
 #! /usr/bin/env python
#coding:utf-8

aliquot = []

for n in range(1,100):
    if n%3 == 0:
        aliquot.append(n)

print aliquot
```

[illegible]

```
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45,
48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90,
93, 96, 99]
```

□□□□□□□□ for □□□ if □□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

```
>>> range(3,100,3)
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45,
48, 51, 54, 57, 60, 63, 66, 69, 72, 75, 78, 81, 84, 87, 90,
93, 96, 99]
```

□□□□ for □□□

□□□□□□□□□□□□□□ for □□□□□□□

```
>>> name_str = "qiwsir"
>>> for i in name_str:  #□□□ str □□ for □□
...     print i,
...
q i w s i r

>>> name_list = list(name_str)
>>> name_list
['q', 'i', 'w', 's', 'i', 'r']
>>> for i in name_list:     #□ list □□□
...     print i,
...
q i w s i r

>>> name_set = set(name_str)    #set □□□□
>>> name_set
set(['q', 'i', 's', 'r', 'w'])
>>> for i in name_set:
...     print i,
...
q i s r w

>>> name_tuple = tuple(name_str)
>>> name_tuple
('q', 'i', 'w', 's', 'i', 'r')
>>> for i in name_tuple:        #tuple □□□
...     print i,
...
q i w s i r
```

```
>>> name_dict={"name":"qiwsir","lang":"python","website":"qiwsir.github.io"}
>>> for i in name_dict:             #dict [illegible]
...     print i,"-->",name_dict[i]
...
lang --> Python
website --> qiwsir.github.io
name --> qiwsir
```

[illegible] for [illegible]

[illegible]

```
>>> a_dict = {"name":"qiwsir", "lang":"python", "email":"qiwsir@gmail.com", "website":"www.itdiffer.com"}
```

[illegible](2)[illegible] items/iteritems/keys/iterkeys/values/itervalues[illegible]

```
>>> for k in a_dict.keys():
...     print k, a_dict[k]
...
lang python
website www.itdiffer.com
name qiwsir
email qiwsir@gmail.com
```

[illegible]/[illegible]

```
>>> for k,v in a_dict.items():
...     print k,v
...
lang python
website www.itdiffer.com
name qiwsir
email qiwsir@gmail.com

>>> for k,v in a_dict.iteritems():
...     print k,v
...
lang python
website www.itdiffer.com
```

```
name qiwsir
email qiwsir@gmail.com
```

[illegible] `iteritems()` [illegible]

[illegible]

```
>>> for k in a_dict.keys():
...     print k, a_dict[k]
...
lang python
website www.itdiffer.com
name qiwsir
email qiwsir@gmail.com
```

[illegible]

```
>>> for v in a_dict.values():
...     print v
...
python
www.itdiffer.com
qiwsir
qiwsir@gmail.com

>>> for v in a_dict.itervalues():
...     print v
...
python
www.itdiffer.com
qiwsir
qiwsir@gmail.com
```

[illegible] values[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□(4)

for □□□ Python □□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□(2)□□□□□□□“□□□□(iterable)”□□□□□□□□□□□□□□□□□□□“□□”□□□□□
Python □□□□□□□□□

□□□□ Python □□□□□□ for □□□□□□□□□□□□□□□□□□□

□□□□□□□□ for □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□a = [1,2,3,4,5], b = [9,8,7,6,5]□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□

□□□□□□□□□□□□□ computer □□□□□□□□□□□□

□□□□□□□□□□□□□□□□ 5□□□□□□□□□□□□□□□□□□□□□□□□□□□ a[i]+b[i],
(i=0,1,2,3,4)□□□□□□□□□□□□□□□□□□□□□□□□□ for □□□□□□□□

```
>>> a = [1,2,3,4,5]
>>> b = [9,8,7,6,5]
>>> c = []
>>> for i in range(len(a)):
...     c.append(a[i]+b[i])
...
>>> c
[10, 10, 10, 10, 10]
```

□□ for □□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□
`zip()` □□□□□□□□□□□□□□□□□□□

zip □□□□□□□□□□□□□ help(zip),□□□□□□□□

> zip(...) zip(seq1 [, seq2 [...]]) -> [(seq1[0], seq2[0] ...), (...)]
>
> Return a list of tuples, where each tuple contains the i-th element from each of the argument sequences. The returned list is truncated in length to the length of the shortest argument sequence.

seq1, seq2 [illegible]

```
>>> a = "qiwsir"
>>> b = "github"
>>> zip(a,b)
[('q', 'g'), ('i', 'i'), ('w', 't'), ('s', 'h'), ('i',
'u'), ('r', 'b')]
```

[illegible]"the length of the shortest argument sequence"[illegible]

```
>>> c = [1,2,3]
>>> d = [9,8,7,6]
>>> zip(c,d)
[(1, 9), (2, 8), (3, 7)]

>>> m = {"name","lang"}
>>> n = {"qiwsir","python"}
>>> zip(m,n)
[('lang', 'python'), ('name', 'qiwsir')]
```

m[illegible]n [illegible]

```
>>> s = {"name":"qiwsir"}
>>> t = {"lang":"python"}
>>> zip(s,t)
[('name', 'lang')]
```

zip [illegible] list [illegible]

[illegible]

```
>>> a
'qiwsir'
>>> c
[1, 2, 3]
>>> zip(c)
[(1,), (2,), (3,)]
```

```
>>> zip(a)
[('q',), ('i',), ('w',), ('s',), ('i',), ('r',)]
```

[illegible] `zip()` [illegible]

```
>>> d = []
>>> for x,y in zip(a,b):
...     d.append(x+y)
...
>>> d
[10, 10, 10, 10, 10]
```

[illegible]

[illegible]

```
>>> a = [1,2,3,4,5]
>>> b = ["python","www.itdiffer.com","qiwsir"]
```

[illegible]“[illegible]”[illegible]

```
>>> length = len(a) if len(a)<len(b) else len(b)
>>> length
3
```

[illegible] Pythonic [illegible]

```
>>> for i in range(length):
...     c.append(str(a[i]) + ":" + b[i])
...
>>> c
['1:python', '2:www.itdiffer.com', '3:qiwsir']
```

[illegible]“[illegible]”[illegible] `a[i]` [illegible] int [illegible] str [illegible]

[illegible] `zip()` [illegible]

```
>>> d = []
>>> for x,y in zip(a,b):
...     d.append(x + y)
...
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
```

```
TypeError: unsupported operand type(s) for +: 'int' and 
'str'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> for x,y in zip(a,b):
...     d.append(str(x) + ":" + y)
...
>>> d
['1:python', '2:www.itdiffer.com', '3:qiwsir']
```

□□□□□□□□□□

□□□**computer** □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> result
[(2, 11), (4, 13), (6, 15), (8, 17)]
>>> zip(*result)
[(2, 4, 6, 8), (11, 13, 15, 17)]
```

`zip()` □□□□□□□□□□□□□□□

□□□□□□□□□

□□□□□□ dictionary□myinfor = {"name":"qiwsir","site":"qiwsir.github.io","lang":"python"},□□□□□□□□□ infor = {"qiwsir":"name","qiwsir.github.io":"site","python":"lang"}

□□□

□□□□□□□□□ for □□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> infor = {}
>>> for k,v in myinfor.items():
...     infor[v]=k
...
>>> infor
{'python': 'lang', 'qiwsir.github.io': 'site', 'qiwsir': 
'name'}
```

□□□ zip() □□□□

```
>>> dict(zip(myinfor.values(),myinfor.keys()))
{'python': 'lang', 'qiwsir.github.io': 'site', 'qiwsir':
```

```
'name'}
```

□□□□□□□□□□□□□□ zip() □□□

```
>>> myinfor.values()    #□□□□ list
['Python', 'qiwsir', 'qiwsir.github.io']
>>> myinfor.keys()
['lang', 'name', 'site']
>>> temp = zip(myinfor.values(),myinfor.keys())     #□□□□□
list□□□□□□□□ tuple
>>> temp
[('python', 'lang'), ('qiwsir', 'name'),
('qiwsir.github.io', 'site')]

>>> dict(temp)                                  #□□□□ dict() □□□□□□
□□□□□□ dictionary
{'Python': 'lang', 'qiwsir.github.io': 'site', 'qiwsir':
'name'}
```

□□□□□□□□ zip()□□□□□□□□□□□□□□□□□□□□□□

enumerate

□□□□□□□□□□□□□□□□□□□□□ `for i in range(len(list))` □□□□□□□ list □□□□□□□□□□□□ list[i]□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□:

```
>>> for i in range(len(week)):
...     print week[i]+' is '+str(i)     #□□□i □ int □□□□□□□
□□□ + □□□□□□ str □□
...
monday is 0
sunday is 1
friday is 2
```

Python □□□□□□□□□□ enumerate□□□□□□□□□□

```
>>> for (i,day) in enumerate(week):
...     print day+' is '+str(i)
...
monday is 0
```

```
sunday is 1
friday is 2
```

□□□□□□□□□□

> Return an enumerate object. sequence must be a sequence, an iterator, or some other object which supports iteration. The next() method of the iterator returned by enumerate() returns a tuple containing a count (from start which defaults to 0) and the values obtained from iterating over sequence:

□□□□□□□□□□□□□□□□□□□□□□

```
>>> seasons = ['Spring', 'Summer', 'Fall', 'Winter']
>>> list(enumerate(seasons))
[(0, 'Spring'), (1, 'Summer'), (2, 'Fall'), (3, 'Winter')]
>>> list(enumerate(seasons, start=1))
[(1, 'Spring'), (2, 'Summer'), (3, 'Fall'), (4, 'Winter')]
```

□□□□□□□□□

```
>>> mylist = ["qiwsir",703,"python"]
>>> enumerate(mylist)
<enumerate object at 0xb74a63c4>
```

□□□□□□□□ list □□□□□□□□□□□.□□□□□□□

```
>>> list(enumerate(mylist))
[(0, 'qiwsir'), (1, 703), (2, 'python')]
```

□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□"Do you love Canglaoshi? Canglaoshi is a good teacher."□□□"Canglaoshi"□□□"PHP".

□□□

```
>>> raw = "Do you love Canglaoshi? Canglaoshi is a good
teacher."
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `enumerate()` □□□□□□□□□□

```
>>> list(enumerate(raw))
[(0, 'D'), (1, 'o'), (2, ' '), (3, 'y'), (4, 'o'), (5,
'u'), (6, ' '), (7, 'l'), (8, 'o'), (9, 'v'), (10, 'e'),
(11, ' '), (12, 'C'), (13, 'a'), (14, 'n'), (15, 'g'), (16,
'l'), (17, 'a'), (18, 'o'), (19, 's'), (20, 'h'), (21,
'i'), (22, '?'), (23, ' '), (24, 'C'), (25, 'a'), (26,
'n'), (27, 'g'), (28, 'l'), (29, 'a'), (30, 'o'), (31,
's'), (32, 'h'), (33, 'i'), (34, ' '), (35, 'i'), (36,
's'), (37, ' '), (38, 'a'), (39, ' '), (40, 'g'), (41,
'o'), (42, 'o'), (43, 'd'), (44, ' '), (45, 't'), (46,
'e'), (47, 'a'), (48, 'c'), (49, 'h'), (50, 'e'), (51,
'r'), (52, '.')]
```

□□□□□□□□□□□□□ raw □□□□□□

```
>>> raw_lst = raw.split(" ")
```

□□□ `enumerate()`

```
>>> for i, string in enumerate(raw_lst):
...     if string == "Canglaoshi":
...         raw_lst[i] = "PHP"
...
```

□□□□□□□□□□□□□□□ raw_lst □□□□□□□□□"Canglaoshi"□□□"PHP"□□

```
>>> raw lst
['Do', 'you', 'love', 'Canglaoshi?', 'PHP', 'is', 'a',
'good', 'teacher.']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□'Canglaoshi?',□□□□□□
□"Canglaoshi"□□□□

□□□□□□□□□□□

```
>>> for i, string in enumerate(raw_lst):
...     if "Canglaoshi" in string:
...         raw_lst[i] = "PHP"
...
>>> raw lst
['Do', 'you', 'love', 'PHP', 'PHP', 'is', 'a', 'good',
'teacher.']
```

□□□□□□□□□□□□□□□□□□□□□□

list □□

[illegible] 1 [illegible] 9 [illegible] list [illegible]

```
>>> power2 = []
>>> for i in range(1,10):
...     power2.append(i*i)
...
>>> power2
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Python [illegible] list [illegible]

```
>>> squares = [x**2 for x in range(1,10)]
>>> squares
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

[illegible] Python[illegible] Python[illegible]

[illegible] list [illegible]

> List comprehensions provide a concise way to create lists. Common applications are to make new lists where each element is the result of some operations applied to each member of another sequence or iterable, or to create a subsequence of those elements that satisfy a certain condition.

[illegible] Python [illegible]

[illegible] list[illegible] list [illegible]

```
>>> mybag = [' glass',' apple','green leaf ']   #[illegible]
>>> [one.strip() for one in mybag]              #[illegible]
['glass', 'apple', 'green leaf']
```

[illegible] list [illegible]

[illegible]list [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

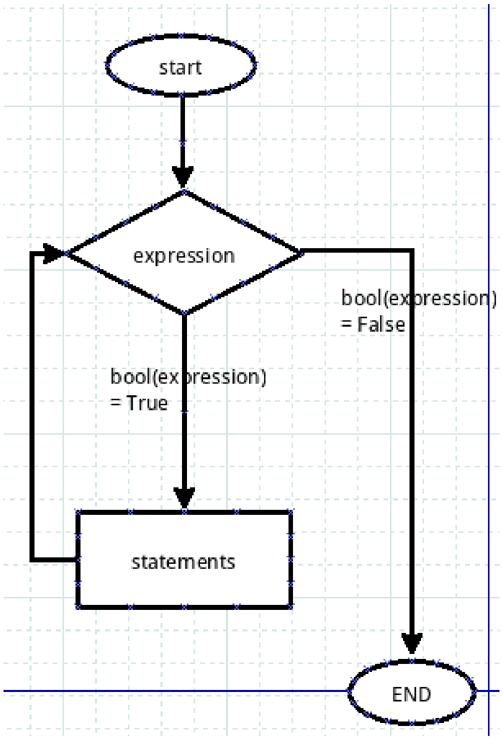
□□(5)

while□□□□□□□“□...□□□”□□□□□□□□□□□□□□□□□□□□□while ... someone do somthing□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□“□...□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□

```
while □□□□ 60 □□-------->□□□□□ 60 □□□□
    □□            -------->□□□□□□□□□□□□□□
```

□□ 60,□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 50,□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ while □□□□□□□□□□□□□□□□□□□


start
expression
bool(expression)
= True
bool(expression)
= False
statements
END

□□□□□□

□□

□□□

```
#! /usr/bin/env Python
#coding:UTF-8

import random

i=0
while i < 4:
    print'********************************'
    num = input('□□□□ 0 □ 9 □□□□□')       #□□□□□□ Python3

    xnum = random.randint(0,9)

    x = 3 - i

    if num == xnum:
        print'□□□□□□□□□□'
        break
    elif num > xnum:
        print'''□□□□!\n □□,□□□□□:%s\n □□□ %s □□□□''' %
(xnum,x)
    elif num < xnum:
        print'''□□□□!\n □□,□□□□□:%s\n □□□ %s □□□□''' %
(xnum,x)
    print'********************************'

    i += 1
```

□□□□□□□□□□□□□□□□□ while i<4□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ while □□□□□□□□□□□i +=1□□□□□□□□□□□□□□□ i □□ 1,□ bool(i<4) □ False □□□□□□□□□□□

□ bool(i<4) □ True □□□ if □□□□□□□□□□□

□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□ 1 □ 100 □□□□

```
num input = raw input("please input one integer that is in
1 to 100:")    #□□□□ Python2.7□□□□□□□□□□□□□
```

[illegible] num_input [illegible]

[illegible]

[illegible] 1 [illegible] 100 [illegible]

1. [illegible]
2. [illegible] 1 [illegible] 100 [illegible]

[illegible]

```
if not num_input.isdigit():     #str.isdigit()[illegible]
    print "Please input interger."
elif int(num_input)<0 and int(num_input)>=100:
    print "The number should be in 1 to 100."
else:
    pass       #[illegible] pass[illegible]
```

[illegible]

```
import random

number = random.randint(1,100)

while True:             #[illegible]
    ...
```

[illegible] num>xnum [illegible] int [illegible] str [illegible]

[illegible]

```
#!/usr/bin/env python
#coding:utf-8

import random

number = random.randint(1,101)

guess = 0
```

```
while True:

    num_input = raw_input("please input one integer that is in 1 to 100:")
    guess += 1

    if not num_input.isdigit():
        print "Please input interger."
    elif int(num_input) < 0 or int(num_input) >= 100:
        print "The number should be in 1 to 100."
    else:
        if number == int(num_input):
            print "OK, you are good.It is only %d, then you successed." % guess
            break
        elif number > int(num_input):
            print "your number is more less."
        elif number < int(num_input):
            print "your number is bigger."
        else:
            print "There is something bad, I will not work"
```

□□□□□□□□□□□□□□

break □ continue

break,□□

```
#!/usr/bin/env python
#coding:utf-8

a = 8
while a:
    if a%2 == 0:
        break
    else:
        print "%d is odd number"%a
        a = 0
print "%d is even number"%a
```

a=8 □□□□□□□□□□□ break□□□□□□□□□□□□□□□□□□□□□

```
8 is even number
```

[illegible] a=9[illegible] else [illegible] print[illegible] a=0[illegible] break [illegible]

```
9 is odd number
0 is even number
```

[illegible] continue [illegible] continue [illegible]

```
#!/usr/bin/env python
#coding:utf-8

a = 9
while a:
    if a%2==0:
        a -=1
        continue    #[illegible]
    else:
        print "%d is odd number"%a #[illegible]
        a -=1
```

[illegible]

while...else

[illegible] if ... else[illegible] else [illegible] while [illegible]

```
#!/usr/bin/env Python

count = 0
while count < 5:
    print count, " is  less than 5"
    count = count + 1
else:
    print count, " is not less than 5"
```

[illegible]

```
0 is less than 5
1 is less than 5
```

```
2 is less than 5
3 is less than 5
4 is less than 5
5 is not less than 5
```

for...else

[illegible] while...else [illegible] for...else[illegible]

```
#!/usr/bin/env python
# coding=utf-8

from math import sqrt

for n in range(99, 1, -1):
    root = sqrt(n)
    if root == int(root):
        print n
        break

else:
    print "Nothing."
```

[illegible]

> [illegible]
>
> [illegible]

[illegible] `for n in range(99, 1, -1)` [illegible] `for n in range(99, 81, -1)` [illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

[illegible](1)

[illegible] computer [illegible] Python [illegible] linux [illegible]

[illegible]

```
>>> dir(file)
['__class__', '__delattr__', '__doc__', '__enter__',
'__exit__', '__format__', '__getattribute__', '__hash__',
'__init__', '__iter__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__setattr__', '__sizeof__',
'__str__', '__subclasshook__', 'close', 'closed',
'encoding', 'errors', 'fileno', 'flush', 'isatty', 'mode',
'name', 'newlines', 'next', 'read', 'readinto', 'readline',
'readlines', 'seek', 'softspace', 'tell', 'truncate',
'write', 'writelines', 'xreadlines']
```

[illegible]

[illegible] `__iter__` [illegible](iterable)[illegible] for [illegible]

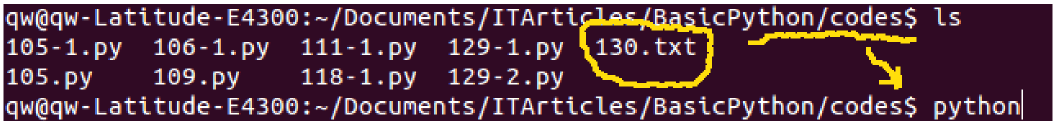
[illegible]

[illegible]130.txt[illegible]

```
learn python
http://qiwsir.github.io
qiwsir@gmail.com
```

[illegible]

[illegible]



```
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ ls
105-1.py  106-1.py  111-1.py  129-1.py  130.txt
105.py    109.py    118-1.py  129-2.py
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ python
```

[illegible] Python[illegible] Python [illegible]

```
>>> f = open("130.txt")      #[illegible]
>>> for line in f:
...     print line
...
learn Python

http://qiwsir.github.io

qiwsir@gmail.com
```

[illegible] Python [illegible] `open("130.txt")` [illegible] 130.txt [illegible] python [illegible]

[illegible] f[illegible] f [illegible] 130.txt [illegible]

[illegible] for [illegible] list[illegible]str[illegible]tuple[illegible] line[illegible]for [illegible] \n [illegible]

[illegible]

[illegible] \n[illegible] for [illegible]print line [illegible] line [illegible] line [illegible] \n[illegible] \n[illegible]\n\n[illegible]

```
>>> f = open('130.txt')
>>> for line in f:
...     print line,     #[illegible] \n [illegible]
...
learn Python
http://qiwsir.github.io
qiwsir@gmail.com
```

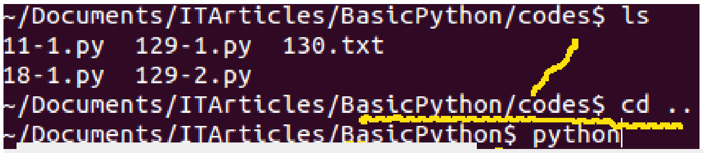
[illegible]

```
>>> f = open('130.txt')
>>> for line in f:
...     print line,
...
learn Python
http://qiwsir.github.io
qiwsir@gmail.com

>>> for line2 in f:     #[illegible] for [illegible]
...     print line2     #[illegible]
...
>>>
```

[illegible] Python [illegible]

[illegible] 130.txt [illegible] 130.txt[illegible]

[illegible]~/Documents/ITArticles/BasicPython[illegible] 130.txt [illegible]



```
~/Documents/ITArticles/BasicPython/codes$ ls
11-1.py  129-1.py  130.txt
18-1.py  129-2.py
~/Documents/ITArticles/BasicPython/codes$ cd ..
~/Documents/ITArticles/BasicPython$ python
```

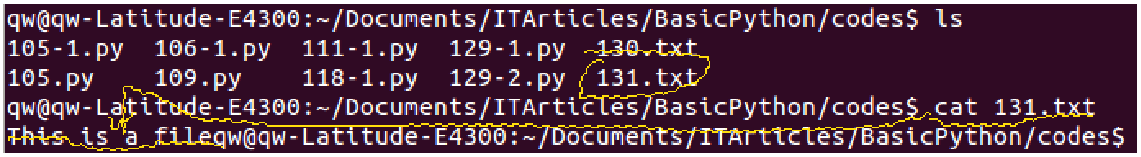
```
>>> f = open("130.txt")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IOError: [Errno 2] No such file or directory: '130.txt'

>>> f = open("./codes/130.txt")     #[illegible]windows[illegible] [illegible]
[illegible] \ [illegible]
>>> for line in f:
...     print line
...
learn Python

http://qiwsir.github.io

qiwsir@gmail.com

>>>
```

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> nf = open("131.txt","w")
>>> nf.write("This is a file")
```

□□□□□□□□□□□□□□□□□□□□□□□□□□



```
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ ls
105-1.py  106-1.py  111-1.py  129-1.py  130.txt
105.py    109.py    118-1.py  129-2.py  131.txt
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ cat 131.txt
This is a fileqw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$
```

□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□ open() □□□□□□□□□□"w"□□□□□□ Python □□□□□□□□□□□□ □□□□□□ open() □□□□□□□□□□□□□□□□□□□□

□□	□□
r	□□□□□□□□□□□□□□□□□
w	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
a	□□□
r+	□□□□□□□□□□□□□□□□□□□□□□
w+	□□□□□□□□□□□□□□□□□□□
a+	□□□□□□□□□□□□□□□□□□□□□□
b	□□□□□□□□□□□□□□□□□□□□□□□□□ Windows □ Dos □□□□ Unix □□□□□□□ □□□□□□□□□

□□□ r □□□□ □□□□□□□□□

```
>>> f = open("130.txt")
>>> f
<open file '130.txt', mode 'r' at 0xb7530230>
>>> f = open("130.txt","r")
>>> f
<open file '130.txt', mode 'r' at 0xb750a700>
```

[illegible]"r"[illegible]

"w":[illegible]

131.txt [illegible]This is a file

```
>>> fp = open("131.txt")
>>> for line in fp:         #[illegible]
...     print line
...
This is a file
>>> fp = open("131.txt","w")    #[illegible]
>>> fp.write("My name is qiwsir.\nMy website is
qiwsir.github.io")  #[illegible]
>>> fp.close()
```

[illegible]

```
$ cat 131.txt  #cat [illegible] linux [illegible] 131.txt [illegible]
My name is qiwsir.
My website is qiwsir.github.io
```

"a":[illegible]

```
>>> fp = open("131.txt","a")
>>> fp.write("\nAha,I like program\n")    #[illegible]
>>> fp.close()                              #[illegible]
[illegible]
```

[illegible]

```
$ cat 131.txt
My name is qiwsir.
My website is qiwsir.github.io
Aha,I like program
```

[illegible]

[illegible] with

[illegible]`file.close()`[illegible]

[illegible]

```
>>> with open("130.txt","a") as f:
...     f.write("\nThis is about 'with...as...'")
...
>>> with open("130.txt","r") as f:
...     print f.read()
...
learn python
http://qiwsir.github.io
qiwsir@gmail.com
hello

This is about 'with...as...'
>>>
```

[illegible] close()[illegible] Python [illegible] Pythonic [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible](2)

[illegible]

[illegible]

[illegible] os [illegible]

```
 >>> import os
>>> file stat = os.stat("131.txt")      #[illegible]
>>> file stat                           #[illegible]
[illegible]
posix.stat result(st mode=33204, st ino=5772566L,
st dev=2049L, st nlink=1, st uid=1000, st gid=1000,
st size=69L, st atime=1407897031, st_mtime=1407734600,
st_ctime=1407734600)

>>> file stat.st ctime                  #[illegible]
1407734600.0882277
```

[illegible] Python [illegible] `time` [illegible]

```
 >>> import time
>>> time.localtime(file stat.st ctime)  #[illegible]
time.struct time(tm year=2014, tm mon=8, tm mday=11,
tm hour=13, tm_min=23, tm_sec=20, tm_wday=0, tm_yday=223,
tm_isdst=0)
```

read/readline/readlines

[illegible] `dir(file)` [illegible] read/readline/readlines[illegible]

[illegible]

[illegible]

[illegible]

```
>>> help(file.read)
```

[illegible]

```
read(...)
    read([size]) -> read at most size bytes, returned as a
string.

    If the size argument is negative or omitted, read until
EOF is reached.
    Notice that when in non-blocking mode, less data than
what was requested
    may be returned, even if no size parameter was given.

readline(...)
    readline([size]) -> next line from the file, as a
string.

    Retain newline.  A non-negative size argument limits
the maximum
    number of bytes to return (an incomplete line may be
returned then).
    Return an empty string at EOF.

readlines(...)
    readlines([size]) -> list of strings, each a line from
the file.

    Call readline() repeatedly and return a list of the
lines so read.
    The optional size argument, if given, is an approximate
bound on the
    total number of bytes in the lines returned.
```

[illegible]

EOF [illegible]End-of-file[illegible]

```
In computing, End Of File (commonly abbreviated EOF[1]) is
a condition in a computer operating system where no more
data can be read from a data source. The data source is
usually called a file or stream. In general, the EOF is
either determined when the reader returns null as seen in
Java's BufferedReader,[2] or sometimes people will manually
insert an EOF character of their choosing to signal when
the file has ended.
```

□□ EOF □□□□□□□□

- read□□□□□□□ size□□“□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“non-blocking”□□□□□□□□□□□□□□□□□□□□□□□□□
- readline□□□□□□ size □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ size□□□□□□□□□□□□□□□□□□□□□□(EOF)□
- readlines□size □□□□□□□□□□□□□□□□□□□□□ `readline()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□you.md□□□□□□□□□□□

> You Raise Me Up When I am down and, oh my soul, so weary; When troubles come and my heart burdened be; Then, I am still and wait here in the silence, Until you come and sit awhile with me. You raise me up, so I can stand on mountains; You raise me up, to walk on stormy seas; I am strong, when I am on your shoulders; You raise me up: To more than I can be.

□□□□□□□□□□□□□□□

```
>>> f = open("you.md")
>>> content = f.read()
>>> content
'You Raise Me Up\nWhen I am down and, oh my soul, so
weary;\nWhen troubles come and my heart burdened be;\nThen,
I am still and wait here in the silence,\nUntil you come
and sit awhile with me.\nYou raise me up, so I can stand on
mountains;\nYou raise me up, to walk on stormy seas;\nI am
strong, when I am on your shoulders;\nYou raise me up: To
more than I can be.\n'
>>> f.close()
```

□□
□□□□□□□□□□□□□□□□□□□□□□□

> □□□□ Python □□'\n'□□□□□□□□ UNIX □□□□□□□□□□□□□ windows □□□'\r\n'□□
> □□□□Python □□□□□□□□□□□□□□□'\r\n'□□□'\n'□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `\n` □□□□□□□□□□□ print □□
□□□□□□□□□□□□□ `\n` □□□□

```
>>> print content
You Raise Me Up
When I am down and, oh my soul, so weary;
When troubles come and my heart burdened be;
Then, I am still and wait here in the silence,
Until you come and sit awhile with me.
You raise me up, so I can stand on mountains;
You raise me up, to walk on stormy seas;
I am strong, when I am on your shoulders;
You raise me up: To more than I can be.
```

□ `readline()` □□□□□□□

```
>>> f = open("you.md")
>>> f.readline()
'You Raise Me Up\n'
>>> f.readline()
'When I am down and, oh my soul, so weary;\n'
>>> f.readline()
'When troubles come and my heart burdened be;\n'
>>> f.close()
```

□□□□□□□□□□□□□ `f.readline()` □□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

f = open("you.md")

while True:
    line = f.readline()
    if not line:         #□ EOF□□□□□□□□□□□□□
        break
    print line ,         #□□□□□□□□□□ print □□□□□ '\n'□□□□□□
□□□□
```

```
f.close()                    #[illegible]
```

[illegible]"you.md"[illegible] 12701.py[illegible] `Python 12701.py` [illegible]

```
~/Documents$ python 12701.py
You Raise Me Up
When I am down and, oh my soul, so weary;
When troubles come and my heart burdened be;
Then, I am still and wait here in the silence,
Until you come and sit awhile with me.
You raise me up, so I can stand on mountains;
You raise me up, to walk on stormy seas;
I am strong, when I am on your shoulders;
You raise me up: To more than I can be.
```

[illegible] `readlines()` [illegible]

```
>>> f = open("you.md")
>>> content = f.readlines()
>>> content
['You Raise Me Up\n', 'When I am down and, oh my soul, so
weary;\n', 'When troubles come and my heart burdened
be;\n', 'Then, I am still and wait here in the silence,\n',
'Until you come and sit awhile with me.\n', 'You raise me
up, so I can stand on mountains;\n', 'You raise me up, to
walk on stormy seas;\n', 'I am strong, when I am on your
shoulders;\n', 'You raise me up: To more than I can be.\n']
```

[illegible] for [illegible]

```
>>> for line in content:
...     print line ,
...
You Raise Me Up
When I am down and, oh my soul, so weary;
When troubles come and my heart burdened be;
Then, I am still and wait here in the silence,
Until you come and sit awhile with me.
You raise me up, so I can stand on mountains;
You raise me up, to walk on stormy seas;
I am strong, when I am on your shoulders;
```

```
You raise me up: To more than I can be.
>>> f.close()
```

□□□□□□

□□□□□□□□□□□□□□□□□□□□ `read()` □□ `readlines()` □□□□□□□□□□□□□□□□□□ while □□□ `readlin()` □□□□□□□□

□□□□□□□□□□fileinput □□

```
>>> import fileinput
>>> for line in fileinput.input("you.md"):
...     print line ,
...
You Raise Me Up
When I am down and, oh my soul, so weary;
When troubles come and my heart burdened be;
Then, I am still and wait here in the silence,
Until you come and sit awhile with me.
You raise me up, so I can stand on mountains;
You raise me up, to walk on stormy seas;
I am strong, when I am on your shoulders;
You raise me up: To more than I can be.
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□ for□

□□□□□□□□□□□□□□□□□□□□□□□□□□□ `dir()` □ `help()` □□□□□□

□□□□□□□□□□□□□

```
>>> for line in f:
...     print line ,
...
You Raise Me Up
When I am down and, oh my soul, so weary;
When troubles come and my heart burdened be;
Then, I am still and wait here in the silence,
Until you come and sit awhile with me.
You raise me up, so I can stand on mountains;
You raise me up, to walk on stormy seas;
I am strong, when I am on your shoulders;
You raise me up: To more than I can be.
```

□□□□□□□□□□□ file □□□□□□□□□□□□□ for □□□□□□

seek

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> f = open("you.md")
>>> f.readline()
'You Raise Me Up\n'
>>> f.readline()
'When I am down and, oh my soul, so weary;\n'
```

□□□□□□□□□□□□□□□ `seek()` □□□□

```
>>> f.seek(0)
```

□□□□□□□□□□□□□□□□□□ `f.readline()` □□□□□□□□

```
>>> f.readline()
'You Raise Me Up\n'
```

□□□□□□□□□□□□□□□□□□□ `tell()` □□□□□

```
>>> f.tell()
17L
>>> f.seek(4)
```

`f.seek(4)` □□□□□□□□□□□□□□□□□□□□□□□□"You "□□□□"R"□□□□□□

```
>>> f.tell()
4L
```

`tell()` □□□□□□□□□□□□□□ `readline()` □□□□□□□□□□□□□□□□

```
>>> f.readline()
'Raise Me Up\n'
>>> f.close()
```

`seek()` □□□□□□□□□□□□

> seek(...) seek(offset[, whence]) -> None. Move to new file position. Argument offset is a byte count. Optional argument whence defaults to 0 (offset from start of file, offset should be >= 0); other values are 1 (move relative to current position, positive or negative), and 2 (move

> relative to end of file, usually negative, although many platforms allow seeking beyond the end of a file). If the file is opened in text mode, only offsets returned by tell() are legal. Use of other offsets causes undefined behavior. Note that not all file objects are seekable.

whence [illegible]

- [illegible] 0[illegible] offset [illegible] 0 [illegible]
- [illegible] 1 [illegible]offset [illegible]
- [illegible] 2 [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible]

[illegible] X [illegible] loop[illegible]iterate[illegible]traversal [illegible] recursion [illegible] X[illegible] X [illegible]“[illegible]”[illegible]“[illegible]”[illegible] X [illegible] X [illegible]

[illegible]

- [illegible]loop[illegible]while [illegible]
- [illegible]iterate[illegible]for [illegible]
- [illegible]recursion[illegible]
- [illegible]traversal[illegible]

[illegible]——[illegible]loop[illegible]iterate[illegible] google[illegible] python [illegible]

[illegible]

[illegible]

[illegible] python [illegible] list[illegible]

```
>>> lst
['q', 'i', 'w', 's', 'i', 'r']
>>> for i in lst:
...     print i,
...
q i w s i r
```

[illegible]

```
>>> lst_iter = iter(lst)    #[illegible] list [illegible] iter()
>>> lst_iter.next()         #[illegible]
'q'
>>> lst_iter.next()
```

```
'i'
>>> lst_iter.next()
'w'
>>> lst_iter.next()
's'
>>> lst_iter.next()
'i'
>>> lst_iter.next()
'r'
>>> lst iter.next()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
StopIteration
```

`iter()` [illegible]

[illegible] `next()` [illegible]“[illegible]”[illegible]

```
>>> while True:
...     print lst_iter.next()
...
Traceback (most recent call last):      #[illegible]
[illegible]
  File "<stdin>", line 2, in <module>
StopIteration
```

[illegible]

```
>>> lst iter = iter(lst)                #[illegible]
>>> while True:
...     print lst_iter.next()
...
q                                       #[illegible]
i
w
s
i
r
Traceback (most recent call last):      #[illegible]
  File "<stdin>", line 2, in <module>
StopIteration
```

[illegible]iter(),[illegible]

> iter(o[, sentinel])
>
> Return an iterator object. The first argument is interpreted very differently depending on the presence of the second argument. Without a second argument, o must be a collection object which supports the iteration protocol (the **iter**() method), or it must support the sequence protocol (the **getitem**() method with integer arguments starting at 0). If it does not support either of those protocols, TypeError is raised. If the second argument, sentinel, is given, then o must be a callable object. The iterator created in this case will call o with no arguments for each call to its next() method; if the value returned is equal to sentinel, StopIteration will be raised, otherwise the value will be returned.

[illegible]...([illegible])

[illegible]

- [illegible]
- [illegible]
- next() [illegible]

[illegible]“[illegible]”[illegible]

[illegible] for [illegible]

[illegible]

[illegible] for [illegible]

[illegible] iter() [illegible] next() [illegible]

[illegible] Python [illegible] iterator[illegible]

[illegible] for [illegible] iter()...next() [illegible] StopIteration [illegible]

[illegible] iter()...next() [illegible] lst_iter [illegible] StopIteration[illegible] for [illegible]

[illegible]

[illegible]208.txt[illegible]

```
Learn python with qiwsir.
There is free python course.
The website is:
http://qiwsir.github.io
Its language is Chinese.
```

[illegible]

```
>>> f = open("208.txt")
>>> f.readline()        #[illegible]
'Learn python with qiwsir.\n'
>>> f.readline()        #[illegible]
'There is free python course.\n'
>>> f.readline()        #[illegible]
'The website is:\n'
>>> f.readline()        #[illegible]
'http://qiwsir.github.io\n'
>>> f.readline()        #[illegible]
'Its language is Chinese.\n'
>>> f.readline()        #[illegible]
''
```

[illegible] readline() [illegible]

```
>>> for line in f:      #[illegible]
...     print line,    #[illegible]
...
```

[illegible]

```
>>> f = open("208.txt")     #[illegible]
>>> for line in f:
...     print line,
...
Learn python with qiwsir.
There is free python course.
The website is:
```

```
http://qiwsir.github.io
Its language is Chinese.
```

□□□□□□□□□□□□□□□□□□□□□ readlines() □□□

□□□□□□ next() □□□□□□□

```
>>> f = open("208.txt")
>>> f.next()
'Learn python with qiwsir.\n'
>>> f.next()
'There is free python course.\n'
>>> f.next()
'The website is:\n'
>>> f.next()
'http://qiwsir.github.io\n'
>>> f.next()
'Its language is Chinese.\n'
>>> f.next()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
StopIteration
```

□□□□ next()□□ iter() □□□□□

□□□□□□□ for □□□□□□□□□□□□□□□□□□□□□ next()□□□□□□□□□□□□□□ for □□□□□□□□□□□□□□□□□□□□□□□□□□ for □□□□□□□□□□□□□□□□

□□

```
>>> [ line for line in open('208.txt') ]
['Learn python with qiwsir.\n', 'There is free python
course.\n', 'The website is:\n',
'http://qiwsir.github.io\n', 'Its language is Chinese.\n']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□

```
>>> list(open('208.txt'))
['Learn python with qiwsir.\n', 'There is free python
course.\n', 'The website is:\n',
'http://qiwsir.github.io\n', 'Its language is Chinese.\n']
```

```
>>> tuple(open('208.txt'))
('Learn python with qiwsir.\n', 'There is free python
course.\n', 'The website is:\n',
'http://qiwsir.github.io\n', 'Its language is Chinese.\n')

>>> "$$$".join(open('208.txt'))
'Learn python with qiwsir.\n$$$There is free python
course.\n$$$The website
is:\n$$$http://qiwsir.github.io\n$$$Its language is
Chinese.\n'

>>> a,b,c,d,e = open("208.txt")
>>> a
'Learn python with qiwsir.\n'
>>> b
'There is free python course.\n'
>>> c
'The website is:\n'
>>> d
'http://qiwsir.github.io\n'
>>> e
'Its language is Chinese.\n'
```

[illegible]

[illegible] for [illegible] iter()...next() [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible]

[illegible] Python [illegible]

[illegible] **1**

[illegible]

[illegible] 10 [illegible][1,2,3,4,5,6,7,8,9,0],[illegible][2,3,4,5,6,7,8,9,0,1]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] 12901.py [illegible]

```
#!/usr/bin/env python
# coding=utf-8

raw = [1,2,3,4,5,6,7,8,9,0]
print raw

b = raw.pop(0)
raw.append(b)
print raw
```

[illegible]

```
$ python 12901.py
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
[2, 3, 4, 5, 6, 7, 8, 9, 0, 1]
```

[illegible] `pop()` [illegible] `append()` [illegible]

[illegible] 2

[illegible]

[illegible]

1. [illegible] 40 [illegible] 0 [illegible] 100 [illegible]
2. [illegible] 40 [illegible]
3. [illegible]

[illegible]

[illegible]

[illegible] 100 [illegible] 1 [illegible] 100 [illegible] 40 [illegible] Python [illegible] Python [illegible]Python [illegible]random[illegible]

```
>>> dir(random)
['BPF', 'LOG4', 'NV MAGICCONST', 'RECIP BPF', 'Random',
'SG MAGICCONST', 'SystemRandom', 'TWOPI', 'WichmannHill',
' BuiltinMethodType', ' MethodType', '  all  ',
'  builtins  ', '  doc  ', '  file  ', '  name  ',
'  package  ', ' acos', ' ceil', ' cos', ' e', ' exp',
' hashlib', ' hexlify', ' inst', ' log', ' pi', ' random',
' sin', ' sqrt', ' test', ' test generator', '_urandom',
' warn', 'betavariate', 'choice', 'division',
'expovariate', 'gammavariate', 'gauss', 'getrandbits',
'getstate', 'jumpahead', 'lognormvariate', 'normalvariate',
'paretovariate', 'randint', 'random', 'randrange',
'sample', 'seed', 'setstate', 'shuffle', 'triangular',
'uniform', 'vonmisesvariate', 'weibullvariate']
```

[illegible] `random.randint()` [illegible]

[illegible]40[illegible] `sum()` [illegible]

[illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

from   future__ import division
import random

score = [random.randint(0,100) for i in range(40)]    #0 [illegible]
100 [illegible] 40 [illegible]
print score

num = len(score)
sum score = sum(score)                #[illegible]
ave num = sum score/num               #[illegible]
less ave = len([i for i in score if i<ave_num])    #[illegible]
[illegible]
print "the average score is:%.1f" % ave num
print "There are %d students less than average." % less_ave

sorted score = sorted(score, reverse=True)    #[illegible]
print sorted_score
```

[illegible] 3

[illegible]

[illegible]"How are you."[illegible]

[illegible]

[illegible](4)[illegible] `strip()` [illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

string = "I love  code."    #[illegible] code [illegible]
print string                #[illegible]
```

```
str_lst = string.split(" ")    #[illegible]
print str_lst

words = [s.strip() for s in str_lst]    #[illegible]
print words

new_string = " ".join(words)    #[illegible]
print new_string
```

[illegible]

```
I love  code.
['I', 'love', '', 'code.']
['I', 'love', '', 'code.']
I love  code.
```

[illegible]

[illegible]

[illegible] `string.split(" ")` [illegible] `strip()` [illegible]

[illegible]

[illegible] `strip()` [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

string = "I love  code."
print string

str_lst = string.split(" ")
print str_lst

words = [s for s in str_lst if s!=""]    #[illegible]
print words

new_string = " ".join(words)
print new_string
```

□□□□□□□□ 12903.py□□□□□□□□□□□

```
 I love  code.
['I', 'love', '', 'code.']
['I', 'love', 'code.']
I love code.
```

OK□□□□□□□□□□□□□ code □□□□□□□□

□□ 4

□□□□

> □□□□□□Donald Ervin Knuth□□□□□□□□□□□□□□The Art of Computer Programming□□1150 □□□□□□ Gopala □□□□□□□□□□□□□□□□□ 1 □ 2 □□□□□□ □□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Leonardo Fibonacci□□□□□□□□□□□□□□□□□□□□
>
> □□□□□□□□□□□□□□;□□□□□□□□□□□□□□□□□□□,□□□□□□□□□□□□□□□□□□□;□□□□□□
>
> □□□□ n □□□□□□□□□□ a □□n+1 □□□□□ b □□□ n+2 □□□□□□ a+b □□ □□□ n+2 □□□□□□□□n+1 □□□ b □□□□□□□□ n+2 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ n □□□□□□ a □

□□□□□□□□□□□□——□□□□□□——□□□□□□□□□□□□□□□□□□□□

```
 a0 = 0                (n=0)
a1 = 1                (n=1)
a[n] = a[n-1] + a[n-2]  (n>=2)
```

□□□□□□□□□□□□□□ n=100 □□□□

□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ for □□□□□□□□□□□□□□□

```
 #!/usr/bin/env Python
# coding=utf-8
```

```
a, b = 0, 1

for i in range(4):    #[illegible]
    a, b = b, a+b

print a
```

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible]

[illegible]Python[illegible]

[illegible]

[illegible]introspection[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]“[illegible]”

[illegible] Python [illegible] Python [illegible]

[illegible] Python [illegible]“[illegible]”[illegible]

[illegible] Python [illegible]“[illegible]”[illegible]

[illegible] Python [illegible]

[illegible]

[illegible] help [illegible] Python [illegible]

```
>>> help()

Welcome to Python 2.7!  This is the online help utility.

If this is your first time using Python, you should
definitely check out
the tutorial on the Internet at
http://docs.python.org/2.7/tutorial/.
```

```
Enter the name of any module, keyword, or topic to get help
on writing
Python programs and using Python modules.  To quit this
help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics,
type "modules",
"keywords", or "topics".  Each module also comes with a
one-line summary
of what it does; to list the modules whose summaries
contain a given word
such as "spam", type "modules spam".

help>
```

[illegible] `keywords`

```
 help> keywords

Here is a list of the Python keywords.  Enter any keyword
to get more help.

and                 elif                if
print
as                  else                import
raise
assert              except              in
return
break               exec                is
try
class               finally             lambda
while
continue            for                 not
with
def                 from                or
yield
del                 global              pass
```

[illegible] Python [illegible] `q` [illegible]

[illegible] Python [illegible] `quit` [illegible]

```
help> quit

You are now leaving help and returning to the Python
interpreter.
If you want to ask for help on a particular object directly
from the
interpreter, you can type "help(object)".  Executing
"help('string')"
has the same effect as typing a particular string at the
help> prompt.
>>>
```

[illegible]

[illegible] Python [illegible] Python [illegible]

[illegible]

[illegible] Python [illegible] .py [illegible] Python [illegible] help('modules') [illegible] help [illegible] modules[illegible] Python [illegible]“[illegible]”[illegible] Python [illegible] Python [illegible]

```
>>> help("modules")

Please wait a moment while I gather a list of all available
modules...
ANSI                _threading_local    gnomekeyring
repr
BaseHTTPServer      _warnings           gobject
requests
MySQLdb             chardet             lsb_release
sre_parse
......([illegible])
PyQt4               codeop              markupbase
stringprep
Queue               collections         marshal
strop
ScrolledText        colorama            math
struct
......([illegible])
Enter any module name to get more help.  Or, type "modules
```

```
spam" to search
for modules whose descriptions contain the word "spam".
```

[illegible] `math`[illegible]`random` [illegible]

[illegible] Python [illegible] `>>>` [illegible] math [illegible] `>>> help("math")` [illegible] `help>` [illegible] `>math` [illegible] math [illegible]

dir()

[illegible]Python [illegible] dir() [illegible]

[illegible]

dir() [illegible] Python [illegible] dir() [illegible]“[illegible]”[illegible] dir() [illegible] keyword [illegible]

```
>>> import keyword
>>> dir(keyword)
['__all__', '__builtins__', '__doc__', '__file__',
'__name__', '__package__', 'iskeyword', 'kwlist', 'main']
```

[illegible] dir() [illegible] keyword [illegible]

```
>>> dir()
['GFileDescriptorBased', 'GInitiallyUnowned',
'GPollableInputStream', 'GPollableOutputStream',
'__builtins__', '__doc__', '__name__', '__package__',
'keyword']
>>> import math
>>> dir()
['GFileDescriptorBased', 'GInitiallyUnowned',
'GPollableInputStream', 'GPollableOutputStream',
'__builtins__', '__doc__', '__name__', '__package__',
'keyword', 'math']
```

dir() [illegible]Python [illegible]

[illegible] dir() [illegible] `__builtins__` [illegible] Python [illegible] `__builtins__` [illegible] Python [illegible]

```
>>>   builtins
<module '__builtin__' (built-in)>
```

[illegible] `__builtins__` [illegible] `__builtin__` [illegible] Python [illegible]

[illegible] `__builtin__.py` [illegible] Python [illegible] dir() [illegible]

```
>>> dir(  builtins  )
['ArithmeticError', 'AssertionError', 'AttributeError',
'BaseException', 'BufferError', 'BytesWarning',
'DeprecationWarning', 'EOFError', 'Ellipsis',
'EnvironmentError', 'Exception', 'False',
'FloatingPointError', 'FutureWarning', 'GeneratorExit',
'IOError', 'ImportError', 'ImportWarning',
'IndentationError', 'IndexError', 'KeyError',
'KeyboardInterrupt', 'LookupError', 'MemoryError',
'NameError', 'None', 'NotImplemented',
'NotImplementedError', 'OSError', 'OverflowError',
'PendingDeprecationWarning', 'ReferenceError',
'RuntimeError', 'RuntimeWarning', 'StandardError',
'StopIteration', 'SyntaxError', 'SyntaxWarning',
'SystemError', 'SystemExit', 'TabError', 'True',
'TypeError', 'UnboundLocalError', 'UnicodeDecodeError',
'UnicodeEncodeError', 'UnicodeError',
'UnicodeTranslateError', 'UnicodeWarning', 'UserWarning',
'ValueError', 'Warning', 'ZeroDivisionError', ' ',
'  debug  ', '  doc  ', '  import  ', '  name  ',
'  package  ', 'abs', 'all', 'any', 'apply', 'ascii',
'basestring', 'bin', 'bool', 'buffer', 'bytearray',
'bytes', 'callable', 'chr', 'classmethod', 'cmp', 'coerce',
'compile', 'complex', 'copyright', 'credits', 'delattr',
'dict', 'dir', 'divmod', 'enumerate', 'eval', 'execfile',
'exit', 'file', 'filter', 'float', 'format', 'frozenset',
'getattr', 'globals', 'hasattr', 'hash', 'help', 'hex',
'id', 'input', 'int', 'intern', 'isinstance', 'issubclass',
'iter', 'len', 'license', 'list', 'locals', 'long', 'map',
'max', 'memoryview', 'min', 'next', 'ngettext', 'object',
'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit',
'range', 'raw input', 'reduce', 'reload', 'repr',
'reversed', 'round', 'set', 'setattr', 'slice', 'sorted',
'staticmethod', 'str', 'sum', 'super', 'tuple', 'type',
'unichr', 'unicode', 'vars', 'xrange', 'zip']
```

dir() [illegible] dir() [illegible] Python [illegible]

```
>>> dir("You raise me up")
['  add  ', '  class  ', '  contains  ', '  delattr__',
'  doc  ', '  eq  ', '  format  ', '  ge  ',
'  getattribute  ', '  getitem  ', '  getnewargs  ',
'  getslice  ', '  gt  ', '  hash  ', '  init  ', '  le__',
'  len  ', '  lt  ', '  mod  ', '  mul  ', '  ne  ',
'  new  ', '  reduce  ', '  reduce ex  ', '  repr  ',
'  rmod  ', '  rmul  ', '  setattr__', '__sizeof__',
'  str  ', '  subclasshook  ',
' formatter field name split', ' formatter parser',
'capitalize', 'center', 'count', 'decode', 'encode',
'endswith', 'expandtabs', 'find', 'format', 'index',
'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace',
'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip',
'partition', 'replace', 'rfind', 'rindex', 'rjust',
'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines',
'startswith', 'strip', 'swapcase', 'title', 'translate',
'upper', 'zfill']
```

[illegible]
`dir(42)`,`dir([])`,`dir(())`,dir({})`,`dir(dir)`[illegible]

[illegible]

[illegible] dir() [illegible] `__doc__` [illegible] Python [illegible] docstring[illegible]

[illegible] `__doc__` [illegible] str [illegible] \n [illegible] Python [illegible] print [illegible]

```
>>> print str.  doc
str(object='') -> string
```

```
Return a nice string representation of the object.
If the argument is a string, the return value is the same object.
```

[illegible] Python [illegible]

[illegible]“[illegible]object[illegible]”[illegible]

[illegible] object [illegible]“[illegible]”[illegible]——[illegible]——[illegible]

[illegible]

[illegible] Python [illegible] dir() [illegible]

[illegible] + [illegible] - [illegible] * [illegible] ** [illegible] / [illegible] % [illegible] < [illegible] > [illegible] Python [illegible]Python [illegible]

[illegible] Python [illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

[illegible] `__name__` [illegible]

```
>>> dir()    #dir() □□
['GFileDescriptorBased', 'GInitiallyUnowned',
'GPollableInputStream', 'GPollableOutputStream',
'  builtins  ', ' _doc__', '__name__', '__package__',
'keyword', 'math']
>>> directory = dir    #□□□
>>> directory()        #□ dir() □□□□□
['GFileDescriptorBased', 'GInitiallyUnowned',
'GPollableInputStream', 'GPollableOutputStream',
'  builtins  ', '  doc  ', '  name__', '__package__',
'directory', 'keyword', 'math']
>>> dir.__name__       #dir() □□□
'dir'
>>> directory.__name__
'dir'

>>>   name__         #□□□□□□
'__main__'
```

□□□□□□□Python □□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□ `__name__` □□□□□□ `' _main__'` □□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□ `  name  ` □□□□□□ `'__main__'` □□□□□□□□□□□□□□□□□□□□□□□□□□ `__name__` □□ □□□□□□□□ Python □□□□□□□□□

```
if   name   == '  main  ':
    # Do something appropriate here, like calling a
    # main() function defined elsewhere in this module.
    main()
else:
    # Do nothing. This module has been imported by another
    # module that wants to make use of the functions,
    # classes and other useful bits it has defined.
```

□□

type() □□ □ types □□□□□□□□□□□□

```
>>> import types
>>> print types.  doc
Define names for all type symbols known in the standard
interpreter.
```

```
Types that are part of optional modules (e.g. array) are
not listed.

>>> dir(types)
['BooleanType', 'BufferType', 'BuiltinFunctionType',
'BuiltinMethodType', 'ClassType', 'CodeType',
'ComplexType', 'DictProxyType', 'DictType',
'DictionaryType', 'EllipsisType', 'FileType', 'FloatType',
'FrameType', 'FunctionType', 'GeneratorType',
'GetSetDescriptorType', 'InstanceType', 'IntType',
'LambdaType', 'ListType', 'LongType',
'MemberDescriptorType', 'MethodType', 'ModuleType',
'NoneType', 'NotImplementedType', 'ObjectType',
'SliceType', 'StringType', 'StringTypes', 'TracebackType',
'TupleType', 'TypeType', 'UnboundMethodType',
'UnicodeType', 'XRangeType', '__builtins__', '__doc__',
'__file__', '__name__', '__package__']
>>> p = "I love Python"
>>> type(p)
<type 'str'>
>>> if type(p) is types.StringType:
...     print "p is a string"
...
p is a string
>>> type(42)
<type 'int'>
>>> type([])
<type 'list'>
>>> type({})
<type 'dict'>
>>> type(dir)
<type 'builtin_function_or_method'>
```

□□

□□□ blist □ clist □□□□□□□□□□□□□□□□□□□□□□□□□ id() □□□

```
>>> print id.  doc
id(object) -> integer

Return the identity of an object.  This is guaranteed to be
unique among
simultaneously existing objects.  (Hint: it's the object's
memory address.)
>>> alist = [1,2,3]
>>> blist = [1,2,3]
>>> clist = blist
>>> id(alist)
2979691052L
>>> id(blist)
2993911916L
>>> id(clist)
2993911916L
>>> alist is blist
False
>>> blist is clist
True
>>> clist.append(4)
>>> clist
[1, 2, 3, 4]
>>> blist
[1, 2, 3, 4]
>>> alist
[1, 2, 3]
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□□□□□□ `dir()` □□ hasattr() □ getattr() □□□□□.

```
>>> print hasattr.  doc
hasattr(object, name) -> bool

Return whether the object has an attribute with the given
name.
(This is done by calling getattr(object, name) and catching
exceptions.)
```

```
>>> print getattr.  doc
getattr(object, name[, default]) -> value

Get a named attribute from an object; getattr(x, 'y') is
equivalent to x.y.
When a default argument is given, it is returned when the
attribute doesn't
exist; without it, an exception is raised in that case.
>>>
>>> hasattr(id, '__doc__')
True

>>> print getattr(id, '__doc__')
id(object) -> integer

Return the identity of an object.  This is guaranteed to be
unique among
simultaneously existing objects.  (Hint: it's the object's
memory address.)
```

□□□

□□□□□□□□□□□□□□□□□□□□□□□□ callable() □□□□□□□□□□□□

```
>>> print callable.  doc__
callable(object) -> bool

Return whether the object is callable (i.e., some kind of
function).
Note that classes are callable, as are instances with a
  call  () method.
>>> callable("a string")
False
>>> callable(dir)
True
```

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible] type() [illegible] isinstance() [illegible]

```
>>> print isinstance.  doc
isinstance(object, class-or-type-or-tuple) -> bool

Return whether an object is an instance of a class or of a
subclass thereof.
With a type as second argument, return whether that is the
object's type.
The form using a tuple, isinstance(x, (A, B, ...)), is a
shortcut for
isinstance(x, A) or isinstance(x, B) or ... (etc.).
>>> isinstance(42, str)
False
>>> isinstance("python", str)
True
```

[illegible]

[illegible]“[illegible]”[illegible]

[illegible]Python [illegible] issubclass() [illegible]

```
>>> print issubclass.  doc _
issubclass(C, B) -> Boolean
Return whether class C is a subclass (i.e., a derived
class) of class B.
>>> class SuperHero(Person):   # SuperHero inherits from
Person...
...     def intro(self):       # but with a new SuperHero
intro
...         """Return an introduction."""
...         return "Hello, I'm SuperHero %s and I'm %s." %
(self.name, self.age)
...
>>> issubclass(SuperHero, Person)
1
>>> issubclass(Person, SuperHero)
0
```

Python [illegible]

[illegible]Python [illegible]

[illegible]

[illegible] word[illegible] vim [illegible]

> [illegible]
>
> [illegible] Barker [illegible] Forward [illegible]Parnas [illegible]
>
> [illegible]

1.[illegible] 2.[illegible] 3.[illegible] 4.[illegible] 5.[illegible]

[illegible] Python [illegible] Python[illegible]Python [illegible] Python [illegible]

[illegible]

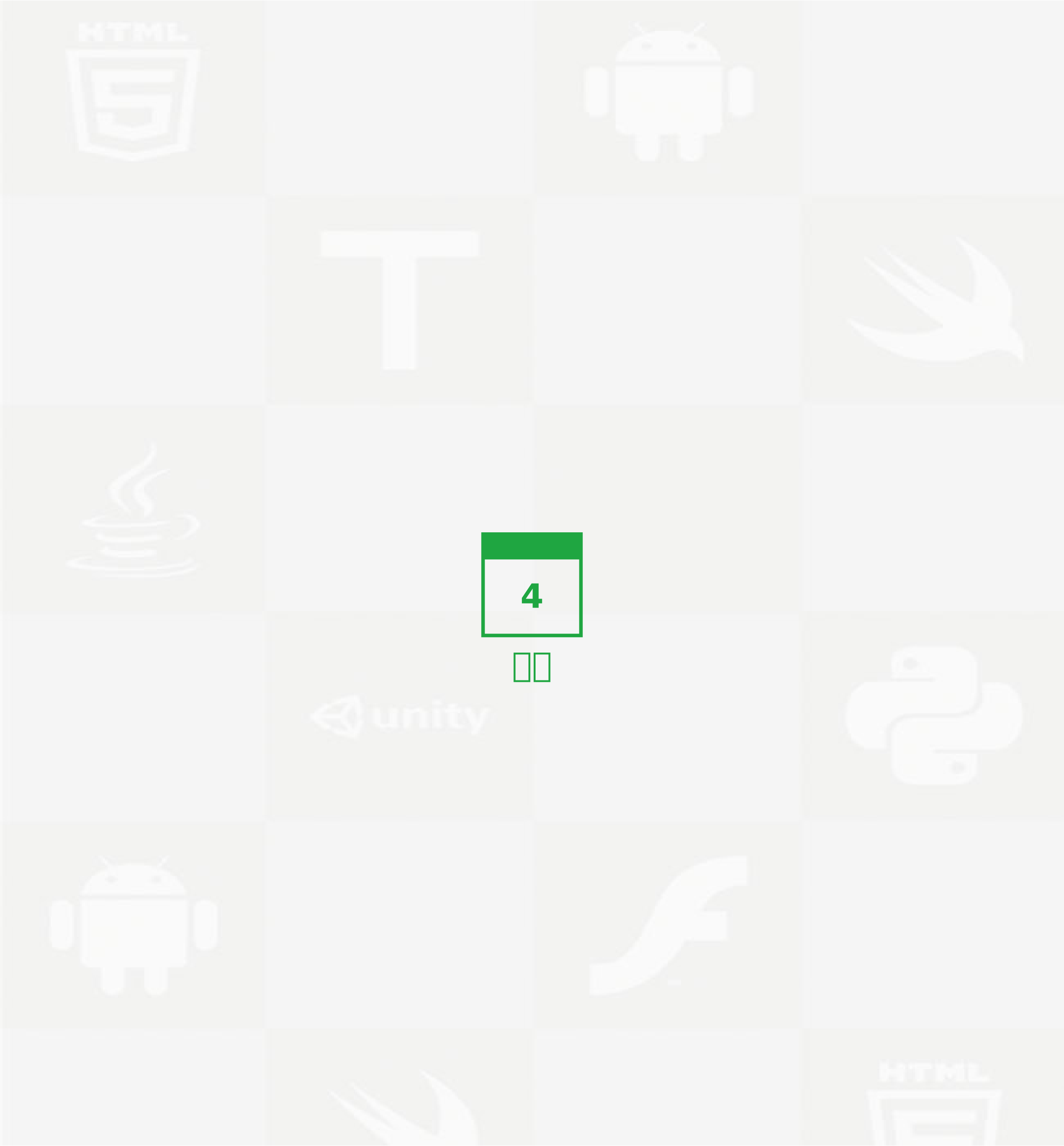
[illegible]

Python [illegible]https://docs.python.org/2/[illegible] Python2.x[illegible] Python3.x [illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]



4

[illegible]

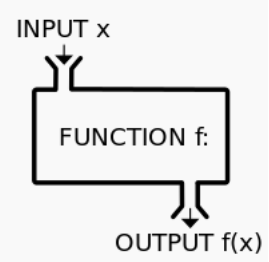
[illegible](1)

[illegible]

[illegible]

> [illegible] 1694 [illegible]
>
> [illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]


INPUT x
FUNCTION f:
OUTPUT f(x)

[illegible]

[illegible] Python [illegible]

[illegible]

[illegible]y=4x+3[illegible]f(x)=4x+3[illegible] x [illegible]

```
[illegible] x=2 [illegible]
f(2) = 4*2+3 = 11
[illegible]f(2) = 11
```

[illegible]

[illegible]

[illegible] x [illegible] x [illegible]4x+3,[illegible] 4 [illegible] 3[illegible] iphone[illegible] 4 [illegible] 3 [illegible] iphone[illegible]

[illegible]x[illegible]

[illegible]**——**[illegible]

[illegible] x[illegible] google [illegible]

[illegible] x[illegible] y,z,k,i,j...[illegible] alpha,beta [illegible]

[illegible]

[illegible] Python [illegible]

[illegible] Python [illegible]

[illegible]alpha,x,j,p_beta[illegible] Python [illegible]

[illegible]

```
>>> a = 2
>>> y=3*a+2
>>> y
8
```

[illegible]

[illegible] a=3,[illegible] y[illegible]

```
>>> a=2
>>> y=3*a+2
>>> y
8
>>> a=3
>>> y
8
```

[illegible]a[illegible]3[illegible] y [illegible] 8[illegible]

[illegible]“[illegible]”[illegible] `a=2` [illegible] 2 [illegible] a [illegible] 8[illegible] y [illegible] 8[illegible] a [illegible]3[illegible]y[illegible] 8[illegible]y [illegible]

```
>>> a=3
>>> y
8
>>> y=3*a+2
>>> y
11
```

[illegible] a=2[illegible]

```
>>> y=3*a+2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  NameError: name 'a' is not defined
```

[illegible]a [illegible]

[illegible]

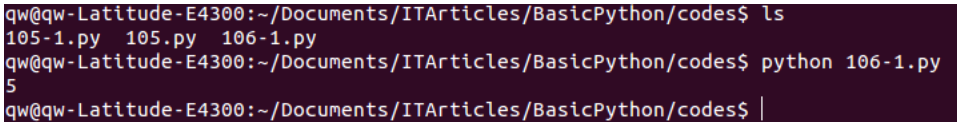
[illegible]“[illegible]”[illegible].py [illegible]

[illegible]

```
 #!/usr/bin/env python
#coding:utf-8

def add function(a,b):
    c = a + b
    print c

if   name   == "  main__":
    add_function(2,3)
```

[illegible] 20101.py[illegible]

[illegible]



```
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ ls
105-1.py  105.py  106-1.py
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ python 106-1.py
5
qw@qw-Latitude-E4300:~/Documents/ITArticles/BasicPython/codes$ 
```

[illegible]

[illegible]

- `def add_function(a,b)` : [illegible] def[illegible]def [illegible] define [illegible]add_function [illegible] Python [illegible] add_function [illegible]“[illegible]”[illegible](a,b)[illegible]
- `c=a+b` [illegible] Python [illegible]([illegible])[illegible] c[illegible]
- `print c` [illegible] c [illegible]
- `if __name__=="__main__"` : [illegible]
- add_function(2,3) [illegible]a=2,b=3[illegible] 2 [illegible] a [illegible]3 [illegible] b [illegible]).

[illegible]

[illegible]

```
def [illegible]([illegible] 1[illegible] 2[illegible]...[illegible] n)[illegible]

    [illegible]
```

[illegible]

[illegible]

- [illegible] Python [illegible]
- def [illegible] define
- [illegible]
- [illegible]
- [illegible] def [illegible] Python [illegible]

[illegible]

```
>>> def name():          #[illegible]
...     print "qiwsir"  #[illegible] 4 [illegible]
...
>>> name()              #[illegible]
qiwsir

>>> def add(x,y):       #[illegible]
...     return x+y      #[illegible] 4 [illegible]
...
>>> add(2,3)            #[illegible] 2+3
5
```

[illegible] add(x,y)[illegible] x,y [illegible] Python [illegible]x,y [illegible] Python [illegible] java [illegible]Python [illegible]

[illegible] add(x,y) [illegible]x,y [illegible]

```
>>> add("qiw","sir")    #[illegible]x="qiw",y="sir"[illegible] x+y,[illegible]"qiw"+"sir"
'qiwsir'

>>> add("qiwsir",4)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 2, in add
TypeError: cannot concatenate 'str' and 'int' objects  #[illegible]
```

[illegible]x+y [illegible] Python [illegible] Python [illegible]**Python** [illegible]

[illegible]

```
>>> result = add(3,4)
>>> result
7
```

[illegible]add(x,y) [illegible] add(x,y) [illegible] add(x,y) [illegible] result [illegible] add(x,y) [illegible] result [illegible]

[illegible]

[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]/[illegible]“[illegible]”[illegible]

[illegible]

Python [illegible] Python [illegible]

- [illegible]:[illegible],[illegible]
- [illegible]:[illegible]myfunction[illegible]my_example_function[illegible]*[illegible]*[illegible]myFunction[illegible]
- [illegible]([illegible] Python [illegible])[illegible]
- [illegible]:[illegible] color = WHITE[illegible]this_is_a_variable = 1[illegible]

[illegible] google [illegible]

- [illegible]
- [illegible]
- [illegible]
- Python[illegible][illegible][illegible],[illegible]

[illegible]

[illegible]

[illegible] Python [illegible]

1. [illegible]“[illegible]”[illegible]
2. [illegible](chong[illegible])[illegible]“[illegible]”[illegible] import [illegible] import ...,[illegible]——[illegible] Python [illegible]——[illegible]

[illegible] add(x,y) [illegible]

```
>>> def add(x,y):       #[illegible]
...     print "x=",x    #[illegible]
...     print "y=",y
...     return x+y
...
>>> add(10,3)           #x=10,y=3
x= 10
y= 3
13

>>> add(3,10)           #x=3,y=10
x= 3
y= 10
13
```

[illegible]

```
>>> add(x=10,y=3)       #[illegible]
x= 10
y= 3
13
```

[illegible]

```
>>> add(y=10,x=3)       #x=3,y=10
x= 3
y= 10
13
```

[illegible]

```
>>> def times(x,y=2):       #y[illegible] 2
...     print "x=",x
...     print "y=",y
...     return x*y
...
>>> times(3)                #x=3,y=2
x= 3
y= 2
6

>>> times(x=3)              #[illegible]
x= 3
y= 2
6
```

[illegible]

```
>>> times(3,4)              #x=3,y=4,y [illegible] 2
x= 3
y= 4
12

>>> times("qiwsir")         #[illegible]
x= qiwsir
y= 2
'qiwsirqiwsir'
```

[illegible] Python [illegible]

[illegible]

[illegible]

1. [illegible]“[illegible]”[illegible]if,while,for [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible] for [illegible] while or range.[illegible]for [illegible]
6. [illegible]
7. [illegible],[illegible] list.append()[illegible]
8. [illegible]
9. [illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□(2)

□□□ `print` □□□

□□□□□□□□□□□□□□□□□□□

□□□

□□□

□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

def fibs(n):
    result = [0,1]
    for i in range(n-2):
        result.append(result[-2] + result[-1])
    return result

if __name__ == "__main__":
    lst = fibs(10)
    print lst
```

□□□□□□□□□□□□□□□□□□□□□□□ 20202.py □□□□□□□□□□□□□□□□□□□□□□□□□ fibs□□□□□□□□□□□□□□□□□□ `lst = fibs(10)` □□□□□□□□□□□□□□ 10□□□□□□□□ n=10 □□□□□□□□

□□□□□□□□

```
$ python 20202.py
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

□□□□□□□ n □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□ fibs □□□□□□□□□□ `return result`□□□□□□□ result □□□□□□□□□□□□□□□□□□□□□□□□□□□□ `lst = fibs(10)` □□□□□□□□□□□□□□□□□ lst □□□□□□ lst □□□ Python □□□□

□□

```
>>> def my_fun():
...     return 1,2,3
...
>>> a = my_fun()
>>> a
(1, 2, 3)
```

□□□□□□□ renturn□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ None□□□□□□□□□□

```
>>> def my_fun():
...     print "I am doing somthin."
...
```

□□

□□ return□

```
>>> a = my_fun()
I am doing somthin.
```

□□□□□□□□□ a□□□□□□

```
>>> print a
None
```

□□□□□□□□□□□□ `return` □□□□□□□□□□□□□ `None` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ None□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> my_fun()
I am doing somthin.
```

□□□□□□ return□□

```
>>> def my_fun():
...     print "I am coding."
...     return
...     print "I finished."
...
>>> my_fun()
I am coding.
```

□□□□□□□□□□□□□□□□□□ print □□□□□□□□□□□□ return□□□□□□ return□□□□□□□□□□□□□□□□□□ print □□□□□□□□□□□□□□□□□□□□□□□□□□ return□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ print □□□□□□□□□□return □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ break □□□□

□□□□□

“□□□□□□□□□□□□□□□□□□□□□□□□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□ `#` □□□Python □□□□□□□Python □□□□□□□□□□□□□□□□□□□□□□□

□□

```
#!/usr/bin/env python
# coding=utf-8

def fibs(n):
    """
    This is a Fibonacci sequence.
    """
    result = [0,1]
    for i in range(n-2):
        result.append(result[-2] + result[-1])
    return result

if __name__ == "__main__":
    lst = fibs(10)
    print lst
```

[illegible]

[illegible] `__doc__` [illegible]

```
>>> def my_fun():
...     """
...     This is my function.
...     """
...     print "I am a craft."
...
>>> my_fun.__doc__
'\n    This is my function.\n    '
```

[illegible] `help(my_fun)` [illegible]

```
Help on function my_fun in module __main__:

my_fun()
    This is my function.
```

[illegible]

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

> [illegible]def [illegible] def [illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] `y = 3x + 2`,[illegible]x[illegible]

[illegible]VB[illegible]

> □□□□□□□□□□ (Visual Basic)
>
> □□
>
> □□ □□□□□□□□□□□□□□□□□□□□□□□□□ □□□
>
> □□□□□□□□□□□□□□□□□□□□□□□
>
> □□□□ Function □ Sub □□□ByVal (Visual Basic) □ ByRef (Visual Basic)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
>
> □□
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
>
> □□ Function □ Sub □□□
>
> □□□

□□

□ Python □□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> def add(x):     #x □□□□□□□□□□
...     a = 10      #a □□□
...     return a+x  #x □□□□□□□□□□□□□□□□□□□□□□□□□□
...
>>> x = 3           #x □□□□□□□□□□□□
>>> add(x)          #□□□ x □□□□□□□□□□□□□ x □□□□ 3
13
>>> add(3)          #□□□□□□□□□
13
```

□□□

□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□ funcx()□□□□□□□□□□□□ x=9□□□□□□□□□□□□□ x=2

```
x = 2

def funcx():
    x = 9
    print "this x is in the funcx:-->",x

funcx()
```

```
print "-------------------------"
print "this x is out of funcx:-->",x
```

□□□□□□□□□□□□□□□□□□□

```
this x is in the funcx:--> 9
--------------------------
this x is out of funcx:--> 2
```

□□□□□□□□ funcx()□□□□ funcx() □□□□□ x=9□□□□□□□□□□□□□□print "this x is out of funcx:-->",x

□□□□□□□□□□□ x □□□□□□□□□□□ x;□□□ x □□□□□□□□□□□ x□□□□□□□□□□□□□□□□□□ x□□□□□□□□□ x □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

```
x = 2
def funcx():
    global x    #□□□□□□□□□□
    x = 9
    print "this x is in the funcx:-->",x

funcx()
print "-------------------------"
print "this x is out of funcx:-->",x
```

□□□□□□□□□□□□□□□□□□□□□□□□ `global x` □□□□□□□□□□□ x □□□□□□□□□□□□ x □□□□□□□□ x □□□□□□□□□ x=9 □ x □□□□□□□□□ 9□□□□□□□□□□□□□

```
this x is in the funcx:--> 9
--------------------------
this x is out of funcx:--> 9
```

□□

□□□□

□□

> □□□□□□□□Namespace□identifier□□□
>
> □□□□ Bill □ X □□□□□□□□□ 123□□ John □ Y □□□□□□□□□□□□□ 123□□
>
> □□“□□□□□□□”□□
>
> □□ C++ □ Python□□□□□□□□□□□□□□□□□□□□□□□□

> [illegible]
>
> [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible] Python [illegible]

[illegible]Python [illegible] dict[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible] vars[illegible]

```
>>> x = 7
>>> scope = vars()
>>> scope['x']
7
>>> scope['x'] += 1
>>> x
8
>>> scope['x']
8
```

[illegible]“[illegible]”[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible](3)

[illegible]πr^2[illegible]

> [illegible]

[illegible]——[illegible]——[illegible]

[illegible]

[illegible]Python [illegible]

```
def func(x,*arg):
    print x         #[illegible] x [illegible]
    result = x
    print arg       #[illegible] *arg [illegible]
    for i in arg:
        result +=i
    return result

print func(1,2,3,4,5,6,7,8,9)    #[illegible]  2[illegible]
```

[illegible]

```
1                        #[illegible] print[illegible]x[illegible] 1
(2, 3, 4, 5, 6, 7, 8, 9) #[illegible] print[illegible] arg [illegible]
45                       #[illegible]
```

[illegible] *arg[illegible] arg [illegible]

- [illegible] 1 [illegible] x
- [illegible] 2,3,4,5,6.7.8.9 [illegible] tuple [illegible] arg

[illegible] *args[illegible] * [illegible]

```
>>> def foo(*args):
...     print args      #[illegible]
...
```

[illegible] *args [illegible]

```
>>> foo(1,2,3)
(1, 2, 3)

>>> foo("qiwsir","qiwsir.github.io","python")
('qiwsir', 'qiwsir.github.io', 'python')
```

```
>>> foo("qiwsir",307,["qiwsir",2],{"name":"qiwsir","lang":"python"})
('qiwsir', 307, ['qiwsir', 2], {'lang': 'python', 'name': 'qiwsir'})
```

□□□□□□□□□□□□□□□ tuple □□

```
>>> foo("python")
('python',)
```

□□□□□□□□□□□□ tuple □□□□□□□□□□□ tuple □□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□ `*args` □□□□□□□□□□□□□

```
>>> def foo(x, *args):
...     print "x:",x
...     print "tuple:",args
...
>>> foo(7)
x: 7
tuple: ()
```

□□□ `*args` □□□□□□□□□□□ tuple□

> □□□□□□□□□□□□□□□□ foo□bar□foobar □□
>
> □□□□□□□□□□□□□□□□□□□□□□□□ foobar □□□□□□□□□□□□□□□□□“□□□”□□□
>
> □□□□□□“foobar”□□□□ 1960 □□□ 1970 □□□□□□□□□□□□□□□□□□□□□□□□□□□“foobar”□□□□□□□□□□□□□ foo □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“1”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“bar”□□□□□□□□□□□□□□□“foo”□□□□“FUBAR”□□□
>
> □□“foobar”□□□□“foo”□“bar”□□□□□□□□□□□□□□□□ Hello World □□□□□□□□□□□□□□□□□□□□□□□□□□□“foo”□□□□□□□□□□□□□□“bar”□□□□□□□□□

□□□ *args □□□□□□□□□□□□□□□□□□□□ **kargs □□□□□□□□□□□□□□□□□□□

```
>>> def foo(**kargs):
...     print kargs
...
>>> foo(a=1,b=2,c=3)    #□□□□□□□□□□□□□□□□□□□
{'a': 1, 'c': 3, 'b': 2}
```

□□□□□□□ foo(1,2,3)□□□□□□□□□□□□□

```
>>> foo(1,2,3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: foo() takes exactly 0 arguments (3 given)
```

□□□ `**kargs` □□□□□□□□□□□ dict □□□□□□□□□□□□□□□□□□□□□□“□”□“□”□□□□□□□□□□□□□□□□□□□

□□□

```
>>> def foo(x,y,z,*args,**kargs):
...     print x
...     print y
...     print z
...     print args
...     print kargs
...
>>> foo('qiwsir',2,"python")
qiwsir
2
python
()
{}
>>> foo(1,2,3,4,5)
1
2
3
(4, 5)
{}
>>> foo(1,2,3,4,5,name="qiwsir")
1
2
3
(4, 5)
{'name': 'qiwsir'}
```

□ good □□□□□□□□□□□□□□□□

□□□□□□□□

```
>>> def add(x,y):
...     return x + y
...
>>> add(2,3)
5
```

□□

```
>>> bars = (2,3)
>>> add(*bars)
5
```

□□□□□□□□□□□□□□ `bars` □□□ `add(*bars)` □□□

```
>>> bars = (2,3,4)
>>> add(*bars)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: add() takes exactly 2 arguments (3 given)
```

□□□□

□□□□□□ `*` □□□□□□□□□□ `**` □□□□□□□□□□□□□□□□

```
>>> def book(author,name):
...     print "%s is writing %s" % (author,name)
...
>>> bars = {"name":"Starter learning Python","author":"Kivi"}
>>> book(**bars)
Kivi is writing Starter learning Python
```

□□

□□

python □□

def foo(p1,p2,p3,...)

□□□

```
>>> def foo(p1,p2,p3):
...     print "p1==>",p1
...     print "p2==>",p2
...     print "p3==>",p3
...
>>> foo("python",1,["qiwsir","github","io"])    #□□□□□□□
p1==> python
p2==> 1
p3==> ['qiwsir', 'github', 'io']

>>> foo("python")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: foo() takes exactly 3 arguments (1 given)    #□□□□□□□

>>> foo("python",1,2,3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: foo() takes exactly 3 arguments (4 given)    #□□ 3 □□□□□□□□□□ 4
□□□□
```

def foo(p1=value1,p2=value2,...)

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> foo(p3=3,p1=10,p2=222)
p1==> 10
p2==> 222
p3==> 3
```

[illegible]

```
>>> def foo(p1,p2=22,p3=33):    #[illegible] p2,p3 [illegible]
...     print "p1==>",p1
...     print "p2==>",p2
...     print "p3==>",p3
...
>>> foo(11)     #p1=11[illegible]
p1==> 11
p2==> 22
p3==> 33
>>> foo(11,222)     #[illegible]p2=222,p3 [illegible]
p1==> 11
p2==> 222
p3==> 33
>>> foo(11,222,333)  #[illegible]
p1==> 11
p2==> 222
p3==> 333

>>> foo(11,p2=122)
p1==> 11
p2==> 122
p3==> 33

>>> foo(p2=122)     #p1 [illegible]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: foo() takes at least 1 argument (1 given)
```

def foo(*args)

[illegible] args [illegible] *[illegible]

```
>>> def foo(*args):         #[illegible]
...     print args
...
>>> foo("qiwsir.github.io") #[illegible] tuple [illegible]
('qiwsir.github.io',)
>>> foo("qiwsir.github.io","python")
('qiwsir.github.io', 'python')
```

def foo(**args)

[illegible] arg=val [illegible]

```
>>> def foo(**args):    #[illegible] dictionary [illegible]
...     print args
...

>>> foo(1,2,3)          #[illegible]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```

```
TypeError: foo() takes exactly 0 arguments (3 given)

>>> foo(a=1,b=2,c=3)    #□□□□□□□□□□□□□□
{'a': 1, 'c': 3, 'b': 2}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> def foo(x,y=2,*targs,**dargs):
...     print "x==>",x
...     print "y==>",y
...     print "targs_tuple==>",targs
...     print "dargs_dict==>",dargs
...

>>> foo("1x")
x==> 1x
y==> 2
targs_tuple==> ()
dargs_dict==> {}

>>> foo("1x","2y")
x==> 1x
y==> 2y
targs_tuple==> ()
dargs_dict==> {}

>>> foo("1x","2y","3t1","3t2")
x==> 1x
y==> 2y
targs_tuple==> ('3t1', '3t2')
dargs_dict==> {}

>>> foo("1x","2y","3t1","3t2",d1="4d1",d2="4d2")
x==> 1x
y==> 2y
targs_tuple==> ('3t1', '3t2')
dargs_dict==> {'d2': '4d2', 'd1': '4d1'}
```

□□□

□□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

□□(4)

□□□□□□□□□□□□□□□□□□□□□□□□ X □□□□□□□□□□□□“□□”□“□□”□“□□”□□□□□□“□□”□

□□

□□□□□□

> □□□□□□.

□□□“□□”□□□□□□□□□□□□□□□□.

> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
> □□“□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□……””

□□□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□)□

> □□□□□□Recursion□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

def fib(n):
    """
    This is Fibonacci by Recursion.
    """
    if n==0:
        return 0
    elif n==1:
        return 1
    else:
        return fib(n-1) + fib(n-2)

if   name   == "__main__":
    f = fib(10)
    print f
```

□□□□□□□□□□□□□□□□□□□ n=10 □□□□□□□

```
$ python 20401.py
55
```

`fib(n-1) + fib(n-2)` □□ n=3□

1. n=3,fib(3)□□□□□ `return fib(3-1) + fib(3-2)` □□

2. □□ fib(3-1),□ fib(2)□□□□ else □□□□□□□ `fib(2-1) + fib(2-2)`
3. fib(2-1) □ fib(1)□□□□□□□□ elif □□□□□ 1□□ fib(2-1)=1□□□□□□□□ fib(2-2)=0□□□□□□□□□□□ □□□□□ `fib(3-1)=1+0=1`
4. □□□ fib(3-2),□□□□□□□□□□□□ 1□□ fib(3-2)=1
5. □□□□□□□□□□□□`fib(3-1) + fib(3-2) = 1 + 1 = 2`□□□□□□ 2 □□□□□

□□□□ fib(3) □□□□ 2□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `a0=0,a1=1` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□

□□□□□□□□□□□□□□□□ `a0=0,a1=1` □□□

```
#!/usr/bin/env Python
# coding=utf-8

"""
the better Fibonacci
"""
meno = {0:0, 1:1}    #□□□

def fib(n):
    if not n in meno:    #□□□□□□□□□□
        meno[n] = fib(n-1) + fib(n-2)
    return meno[n]

if   name   == "__main__":
    f = fib(10)
    print f

#□□□□
$ python 20402.py
55
```

□□□□□□□□□□□□□□ Python □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□“□□□□□”□□ □□

> □□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□”□“□□□□□□”□“□□□□□”□□□□□□□□□□□□□□□□□□“□□□□”□ □□□□□□□□□□□□□□
>
> □□□□□□□□□□□□□Programming paradigm□□□ □□
>
> □□□ □□□□□□□□□□□□□□

> [illegible]“[illegible]”[illegible]“[illegible]”[illegible] Smalltalk [illegible] Java [illegible] Haskell [illegible] Scheme [illegible] Ruby[illegible] Common Lisp[illegible]Python [illegible] Oz[illegible]
>
> [illegible]C++ [illegible] C++ [illegible] C++ [illegible]

[illegible]

[illegible]Python [illegible]

filter[illegible]map[illegible]reduce[illegible]lambda[illegible]yield

[illegible]X[illegible]

[illegible] yield [illegible]

lambda

lambda [illegible]

```
>>> def add(x):     #[illegible] 3,[illegible]
...     x += 3
...     return x
...
>>> numbers = range(10)
>>> numbers
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  #[illegible] list[illegible] 3,[illegible] list [illegible]

>>> new_numbers = []
>>> for i in numbers:
...     new_numbers.append(add(i))  #[illegible] add() [illegible] append [illegible] list [illegible]
...
>>> new_numbers
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

[illegible]add() [illegible]

```
>>> new_numbers = [ i+3 for i in numbers ]
>>> new_numbers
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

[illegible]

[illegible] lambda [illegible] add(x)[illegible]

```
>>> lam = lambda x:x+3
>>> n2 = []
>>> for i in numbers:
...     n2.append(lam(i))
...
```

```
>>> n2
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

□□□ lam □□□□ add(x)□□□□□□□□□□□□ lambda x:x+3 □□□ add(x) □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> g = lambda x,y:x+y  #x+y,□□□□□
>>> g(3,4)
7
>>> (lambda x:x**2)(4)  #□□ 4 □□□
16
```

□□□□□□□□□□□ lambda □□□□□□□□

- □ lambda □□□□□□□
- □□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□

```
lambda arg1, arg2, ...argN : expression using arguments
```

□□□□□□□□□□ lambda □□□□□□□□□□□□ (□□□□□□) □□□□□□□□□□□□□□**lambda** □□□□□□□□□□□□□□□□ □□□□□□□□□□□ **lambda** □□

□ lambda □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ list□□□□□□□□□□□ 1 □□□□□□□□□□□□ □□□□ lambda □□□□□□

```
>>> lamb = [ lambda x:x,lambda x:x**2,lambda x:x**3,lambda x:x**4 ]
>>> for i in lamb:
...     print i(3),
...
3 9 27 81
```

lambda □□□□□□□□□□□□□□□□□□□□□□□□□

map

□□□□□□□□□□□□□ lambda □□□□□□□□□□ map □□□□□□

```
>>> numbers
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]      #□□□□□□□□□ 3

>>> map(add,numbers)                #add(x) □□□□□□□□□□□□□□□□□□□□□□□□
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

>>> map(lambda x: x+3,numbers)      #□ lambda □□□□□
[3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

map() □ Python □□□□□□□□□□□□□□□□

```
map(func,seq)
```

func [illegible]seq [illegible] func [illegible]
func [illegible] list [illegible]

[illegible]map [illegible]

```
>>> items = [1,2,3,4,5]
>>> squared = []
>>> for i in items:
...     squared.append(i**2)
...
>>> squared
[1, 4, 9, 16, 25]

>>> def sqr(x): return x**2
...
>>> map(sqr,items)
[1, 4, 9, 16, 25]

>>> map(lambda x: x**2, items)
[1, 4, 9, 16, 25]

>>> [ x**2 for x in items ]     #[illegible]
[1, 4, 9, 16, 25]
```

[illegible]

[illegible] map() [illegible]

> map(function, iterable, ...)
>
> Apply function to every item of iterable and return a list of the results. If additional iterable arguments are passed, function must take that many arguments and is applied to the items from all iterables in parallel. If one iterable is shorter than another it is assumed to be extended with None items. If function is None, the identity function is assumed; if there are multiple arguments, map() returns a list consisting of tuples containing the corresponding items from all iterables (a kind of transpose operation). The iterable arguments may be a sequence or any iterable object; the result is always a list.

[illegible]

- [illegible] iterable [illegible] function [illegible] for [illegible]
- [illegible] list[illegible]
- [illegible] function[illegible]

[illegible]

```
>>> lst1 = [1,2,3,4,5]
>>> lst2 = [6,7,8,9,0]
>>> map(lambda x,y: x+y, lst1,lst2)     #[illegible]
[7, 9, 11, 13, 5]
```

[illegible] for [illegible] for [illegible]

```
>>> lst1 = [1,2,3,4,5]
>>> lst2 = [6,7,8,9,0]
>>> lst3 = [7,8,9,2,1]
>>> map(lambda x,y,z: x+y+z, lst1,lst2,lst3)
[14, 17, 20, 15, 6]
```

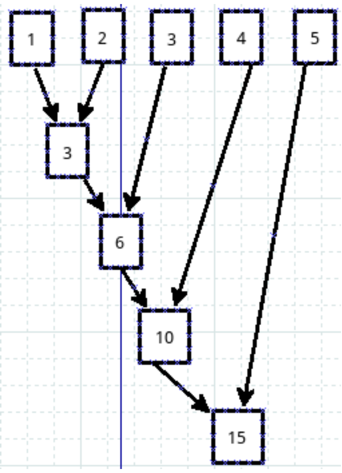
□□□□□ map □□□□□□

reduce

□□□□□□

```
>>> reduce(lambda x,y: x+y,[1,2,3,4,5])
15
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□



□□□ map □□□□□□□□□□□□□□□

```
>>> list1 = [1,2,3,4,5,6,7,8,9]
>>> list2 = [9,8,7,6,5,4,3,2,1]
>>> map(lambda x,y: x+y, list1,list2)
[10, 10, 10, 10, 10, 10, 10, 10, 10]
```

□□□□□□□□□□□□□□□□□□□map□□□□□□reduce □□□□□□□□□□□□

□□□□□□□□□□

> reduce(function, iterable[, initializer])
>
> Apply function of two arguments cumulatively to the items of iterable, from left to right, so as to reduce the iterable to a single value. For example, reduce(lambda x, y: x+y, [1, 2, 3, 4, 5]) calculates ((((1+2)+3)+4)+5). The left argument, x, is the accumulated value and the right argument, y, is the update value from the iterable. If the optional initializer is present, it is placed before the items of the iterable in the calculation, and serves as a default when the iterable is empty. If initializer is not given and iterable contains only one item, the first item is returned. Roughly equivalent to:

```
def reduce(function, iterable, initializer=None):
    it = iter(iterable)
    if initializer is None:
        try:
            initializer = next(it)
        except StopIteration:
            raise TypeError('reduce() of empty sequence with no initial
value')
    accum_value = initializer
    for x in it:
        accum_value = function(accum_value, x)
    return accum_value
```

□□□□□□□□ for □□□□□□ reduce □□□□□□□□□□□

```
>>> lst = range(1,6)
>>> lst
[1, 2, 3, 4, 5]
>>> r = 0
>>> for i in range(len(lst)):
...     r += lst[i]
...
>>> r
15
```

for □□□□reduce □□□□□

□□□□□□□□□□□□□□□□□□ list□a = [3,9,8,5,2],b=[1,4,9,2,6],□□□a[0]*b[0]+a[1]*b[1]+...□□□□

```
>>> a
[3, 9, 8, 5, 2]
>>> b
[1, 4, 9, 2, 6]

>>> zip(a,b)        #□□□□ zip□□□□□□□□□□
[(3, 1), (9, 4), (8, 9), (5, 2), (2, 6)]

>>> sum(x*y for x,y in zip(a,b))    #□□□□□□□

133

>>> new_list = [x*y for x,y in zip(a,b)]    #□□□□□□□□□□□□□□□

>>> #□□□□□□□□□new_tuple = (x*y for x,y in zip(a,b))
>>> new_list
[3, 36, 72, 10, 12]
>>> sum(new_list)     #□□:sum(new_tuple)
133

>>> reduce(lambda sum,(x,y): sum+x*y,zip(a,b),0)    #□□□□□□□□□□□□
133

>>> from operator import add,mul            #□□□□□□□□□□□
>>> reduce(add,map(mul,a,b))
133
```

```
>>> reduce(lambda x,y: x+y, map(lambda x,y: x*y, a,b))  #map,reduce,lambda
[illegible]
133
```

[illegible] Python3[illegible] Python3 [illegible]reduce() [illegible] functools [illegible] `from functools import reduce` [illegible]

filter

filter [illegible]“[illegible]”[illegible] Python [illegible]

> filter(function, iterable)
>
> Construct a list from those elements of iterable for which function returns true. iterable may be either a sequence, a container which supports iteration, or an iterator. If iterable is a string or a tuple, the result also has that type; otherwise it is always a list. If function is None, the identity function is assumed, that is, all elements of iterable that are false are removed.
>
> Note that filter(function, iterable) is equivalent to [item for item in iterable if function(item)] if function is not None and [item for item in iterable if item] if function is None.

[illegible]

[illegible]

```
>>> numbers = range(-5,5)
>>> numbers
[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4]

>>> filter(lambda x: x>0, numbers)
[1, 2, 3, 4]

>>> [x for x in numbers if x>0]     #[illegible]
[1, 2, 3, 4]

>>> filter(lambda c: c!='i', 'qiwsir')  #[illegible]
'qwsr'                                  #“If iterable is a string or a
tuple, the result also has that type;”
```

[illegible] Python [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible]

[illegible]

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible] github [illegible] QQ [illegible]([illegible]:26913719)[illegible]

[illegible]

[illegible]

[illegible]

> [illegible] 2000 [illegible]2000 BC[illegible] 480 [illegible] 300 [illegible]
>
> 7 [illegible]Brahmagupta[illegible]
>
> 11 [illegible] [illegible]·[illegible] Liber embadorum [illegible]([illegible])

[illegible]

```
 #!/usr/bin/env Python
# coding=utf-8

"""
solving a quadratic equation
"""

from   future__ import division
import math

def quadratic equation(a,b,c):
    delta = b*b - 4*a*c
    if delta<0:
        return False
    elif delta==0:
        return -(b/(2*a))
    else:
        sqrt delta = math.sqrt(delta)
        x1 = (-b + sqrt delta)/(2*a)
        x2 = (-b - sqrt_delta)/(2*a)
        return x1, x2
```

```
if __name__ == "__main__":
    print "a quadratic equation: x^2 + 2x + 1 = 0"
    coefficients = (1, 2, 1)
    roots = quadratic_equation(*coefficients)
    if roots:
        print "the result is:",roots
    else:
        print "this equation has no solution."
```

□□□ 20501.py□□□□□□

```
$ python 20501.py
a quadratic equation: x^2 + 2x + 1 = 0
the result is: -1.0
```

□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- □□□□□□□□□□□□□(a)□□□□□ 0□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
- □□□
- □□□Python □□□□□□□□□□□ delta<0□□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□ Python □□□ google □□□□□□□□□□□□□□□□□□□

> Python □□□ Python □□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□ math□□ SymPy(www.sympy.org)□□□□□□ NumPy □□□□□□□□□□

□□□□□□

□□`{"zhangsan":90, "lisi":78, "wangermazi":39}`□□

□□□□

□□□□□□□□□□

- □□□□□□□□□□□□□□□□
- □□□□□□□□□□□□□□□□
- □□

□□□□□□□□□□□□□□□ Python□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□

```
#!/usr/bin/env Python
# coding=utf-8
```

```
"""
□□□□□□
"""
from __future__ import division

def average_score(scores):
    """
    □□□□□.
    """
    score values = scores.values()
    sum scores = sum(score values)
    average = sum scores/len(score_values)
    return average

def sorted_score(scores):
    """
    □□□□□□□□□.
    """
    score lst = [(scores[k],k) for k in scores]
    sort lst = sorted(score lst, reverse=True)
    return [(i[1], i[0]) for i in sort_lst]

def max_score(scores):
    """
    □□□□□□□□□□.
    """
    lst = sorted score(scores)    #□□□□□□□□□ sorted_score
    max score = lst[0][1]
    return [(i[0],i[1]) for i in lst if i[1]==max_score]

def min_score(scores):
    """
    □□□□□□□□□□.
    """
    lst = sorted score(scores)
    min score = lst[len(lst)-1][1]
    return [(i[0],i[1]) for i in lst if i[1]==min_score]

if   name   == "  main  ":
    examine scores = {"google":98, "facebook":99, "baidu":52,
"alibaba":80, "yahoo":49, "IBM":70, "android":76, "apple":99, "amazon":99}

    ave = average score(examine scores)
    print "the average score is: ",ave    #□□□

    sor = sorted score(examine scores)
    print "list of the scores: ",sor      #□□□

    xueba = max score(examine scores)
    print "Xueba is: ",xueba              #□□□

    xuezha = min score(examine scores)
    print "Xuzha is: ",xuezha             #□□□
```

[illegible] 20502.py[illegible]

```
$ python 20502.py
the average score is:  80.2222222222
list of the scores:  [('facebook', 99), ('apple', 99), ('amazon', 99), ('google', 98), ('alibaba', 80), ('android', 76), ('IBM', 70), ('baidu', 52), ('yahoo', 49)]
Xueba is:  [('facebook', 99), ('apple', 99), ('amazon', 99)]
Xuzha is:  [('yahoo', 49)]
```

[illegible]

[illegible]

[illegible] 100 [illegible]

[illegible]

> [illegible]Prime number[illegible]1[illegible] 1 [illegible] 1 [illegible]
>
> [illegible] 1742 [illegible]·[illegible]·[illegible]“[illegible] 2 [illegible]”[illegible] 1973 [illegible]“1+2”[illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

"""
[illegible]
"""

import math

def is_prime(n):
    """
    [illegible]
    """
    if n <= 1:
        return False
    for i in range(2, int(math.sqrt(n) + 1)):
        if n % i == 0:
            return False
    return True

if __name__ == "__main__":
    primes = [i for i in range(2,100) if is_prime(i)]    #[illegible] 2 [illegible] 1 [illegible]
    print primes
```

[illegible]

```
$ python 20503.py
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
71, 73, 79, 83, 89, 97]
```

[illegible] 100 [illegible]

[illegible]

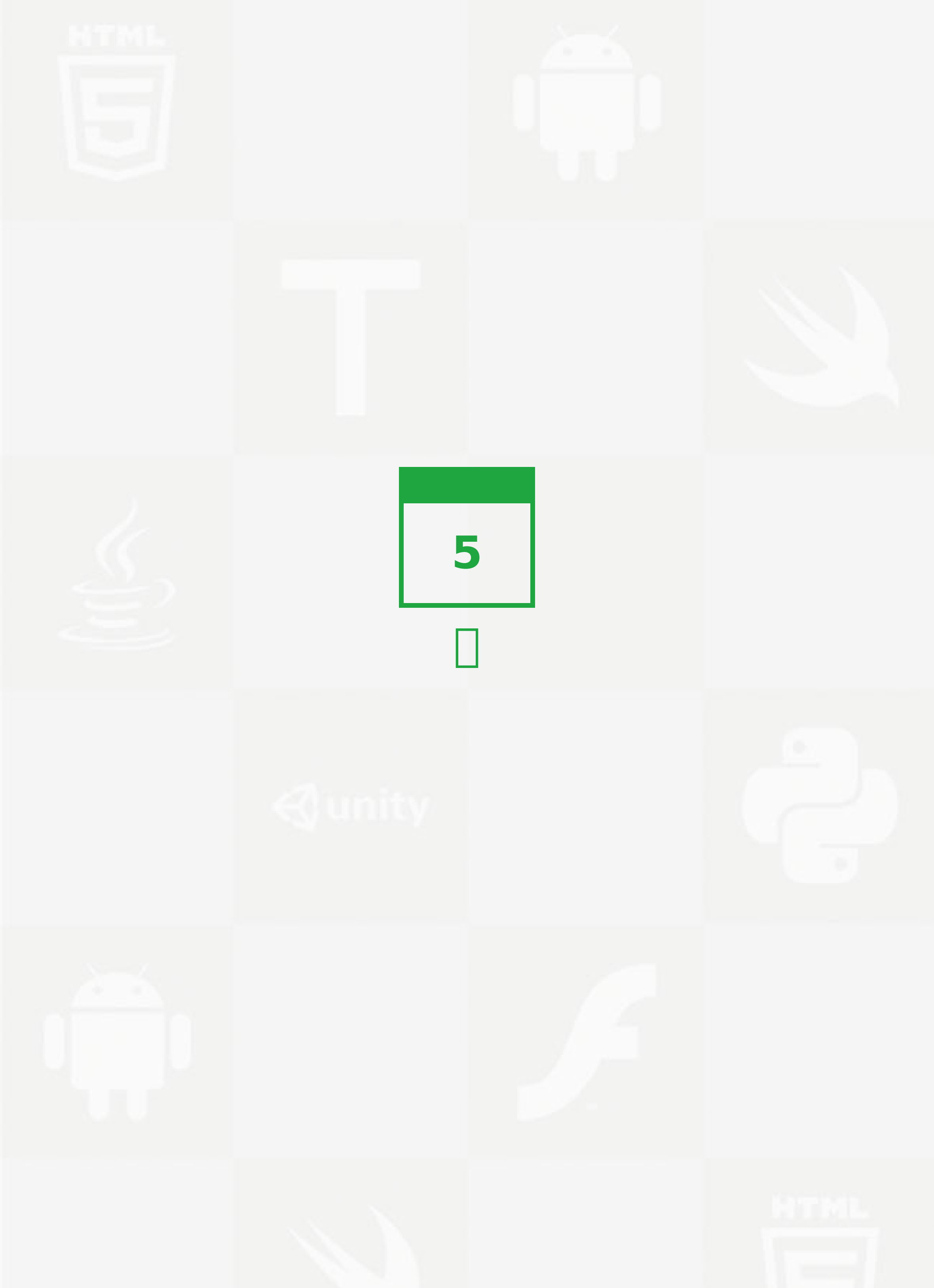
[illegible]

[illegible]

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]



5

[illegible](1)

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible] class[illegible]“[illegible]”[illegible] class [illegible]

[illegible]“[illegible]”[illegible] OOP[illegible]“[illegible]”[illegible]

[illegible] OOP[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

> [illegible]

[illegible]

- [illegible]——[illegible]
- [illegible]——[illegible]
- [illegible]——[illegible]

[illegible]problem space[illegible]

□□

□□□

> □□□object□□□□□□□□□□□□□□□□□□Object Oriented□□□□□□□□□□□
> □□□□□□□Namespace□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□ object □□□“□□”□□□□□□□□□□□□□□□□□□□□“□□”□□□□□“□□”□
□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□Python □□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□”□

□□□□□□□□□□□□□□□□□□ Python □□□□ OOP □□□□□□Python □□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□OOP □□ Grandy Booch □□□□□□□□□□□□□□□□□□□□□□

- □□
- □□□**state**□□□
- □□□**behavior**□□□
- □□□**identity**□□

□□□

□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□

> □□□□□□□□□□□□Object-oriented programming□□□□OOP□□□
> □□
> □□□□□□□□□□□□□□□□□□□□□□□□□□□
>
> □□□
> □□
> □□
> □□□“□□”□□□□□
>
> □□
> □□
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
>
> □□
> □□□□□□□□□□□□□□□□□□□□□□□□□□Object Oriented Design□□
> □ OOD□□□□□□□

□□□□□□□□□□□□□□□□ OOP □□□□

> □□□□□□□□□□□□□□ 1960 □□ Simula □□□□□□□□□□□□□□□□□□□□
> □□
> □□□□□□□□□□□□□□□20 □□ 70 □□□ Smalltalk □□□□□□□□□□□□□
> ——□□□ 30 □□□□□□□□□□□□□□□□□□□□□□□□
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ PDP-1 □□□□□□□□□□□
> □□□□□□□□□□□□capability based architecture□□□□□□□□ 1963 □
> Ivan Sutherland □ Sketchpad □□□□□□□□□□□□□□□□□□□□□□□
> □□ 1960 □□□ Simula 67 □□□□□□□Simula □□□□□□□□-□□·□□□□□
> □□·□□
> □□
> □□

> [illegible]“[illegible]”[illegible]Simula [illegible]“[illegible]”[illegible]——[illegible]
>
> 20 [illegible] 70 [illegible] PARC [illegible] Smalltalk [illegible]Smalltalk [illegible] Simula 67 [illegible] Smalltalk [illegible]——[illegible] Simula [illegible]Smalltalk [illegible] Simula[illegible]Simula 67 [illegible] Lisp[illegible]Pascal[illegible]
>
> [illegible] 80 [illegible] C++——C [illegible]GUI[illegible]GUI [illegible] Mac OS X [illegible]Mac OS X [illegible] Objective-C [illegible] Smalltalk [illegible] C [illegible]GUI [illegible]
>
> [illegible]·[illegible] Modula-2 [illegible] Oberon [illegible]——[illegible] Smalltalk [illegible] C++[illegible]
>
> [illegible]Ada[illegible]BASIC[illegible]Lisp[illegible]Fortran[illegible] Pascal [illegible]“[illegible]”[illegible]Eiffel [illegible]
>
> [illegible]Java [illegible] C [illegible] C++ [illegible] Java [illegible]
>
> [illegible] Python[illegible]Ruby [illegible]
>
> [illegible] UML[illegible]

[illegible] OOP [illegible]

[illegible]

[illegible]

> [illegible]class[illegible]
>
> [illegible]
>
> [illegible]

[illegible] OOP [illegible] OOP[illegible]

[illegible]

[illegible]“[illegible]”——[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

```
class [illegible]:        #[illegible] class [illegible]
    pass
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python□□□□□□□□□□□□□□□
□ Python □□□□□□□□

```
class □□:
    □□ = 90
    □□ = 58
    □□ = 83
    □□ = white
    □□()
    □□()
```

□□□□□□□□“□□”□□.

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□“□□□”□□□□

```
□□□ = □□()
```

□□□□□□□□□□□□□□“□□□”□□□□□□“□□□”□□

```
a = □□□.□□
```

□□□`.`□□□□□□□□□□□□□□□□□□□□ a □□ 90.□□□□□□□□□□□□□□□□□□□□

```
□□□.□□ = black
```

□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□

```
□□□.□□()
```

□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible]

[illegible]qiwsir@126.com,[illegible]

[illegible](2)

[illegible] Python [illegible]

[illegible]

[illegible] Python [illegible] Python2.x [illegible]“[illegible]”[illegible]“[illegible]”[illegible] Python2.2 [illegible]

```
>>> class AA:
...     pass
...
```

[illegible] `AA` [illegible] `pass` [illegible]

[illegible]

```
>>> aa = AA()
```

[illegible]

```
>>> type(AA)
<type 'classobj'>
>>> aa.__class__
<class __main__.AA at 0xb71f017c>
```

```
>>> type(aa)
<type 'instance'>
```

□□□□□□□□□

- `type(AA)`□□□□ `AA` □□□□□□□□`'classobj'`
- `aa.__class__`□aa □□□□□□□□□□□□□□□□□□□ `class__` □□□□□□□□□□□□□□□□□□□□`<class __main__.AA at 0xb71f017c>`□□□□□□□□□□□□□□□□aa □□ AA □□□□□□□ AA □□□□□□□□ `0xb71f017c`□
- `type(aa)`□□□□□□ aa □□□□□□□□□□□`'instance`□□□□□□ □□□□□□□□□□□□

□□□□□□□□□□□□□□□`aa.__class__`□ `type(aa)` □□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> a = 7
>>> a.  class__
<type 'int'>
>>> type(a)
<type 'int'>
```

□□□□□□□□□□□□“□□□□□”□□□□□□□7□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□ Python2.2 □□□□□□□□□

```
>>> class BB(object):
...     pass
...

>>> bb = BB()

>>> bb.  class
<class '__main__.BB'>
```

```
>>> type(bb)
<class '__main__.BB'>
```

[illegible]

[illegible]

[illegible]

[illegible]

```
>>> print aa
<__main__.AA instance at 0xb71efd4c>

>>> print bb
<__main__.BB object at 0xb71efe6c>
```

[illegible]

[illegible]“[illegible]”[illegible]

[illegible] Python3[illegible] Python3 [illegible]

[illegible]

[illegible]

```
>>> class BB(object):
...     pass
...
```

[illegible] `(object)` [illegible]“[illegible]”[illegible] BB [illegible] object [illegible]object [illegible] BB [illegible] object [illegible] Python3 [illegible] object [illegible]

□□□□□□□□□□□□□□□□□“□□”□“□□”□“□□”□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□`__metaclass__ == type`□□□□□□□□□□□□□□□□□□□□`(object)`□□

```
>>> __metaclass__ = type
>>> class CC:
...     pass
...
>>> cc = CC()
>>> cc.__class__
<class '__main__.CC'>
>>> type(cc)
<class '__main__.CC'>
```

□□□□□□□□□□□□□□□□□

□□□

□□□Python □□□□ IDE□□□□□□□□□□ vim□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□ IDE□

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class Person:
    def __init__(self, name):
        self.name = name

    def getName(self):
        return self.name

    def color(self, color):
        print "%s is %s" % (self.name, color)
```

[illegible]“[illegible]”[illegible]

[illegible]

`__metaclass__ = type`[illegible]

[illegible]

`class Person`[illegible]"Person"[illegible] `class HotPerson`[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]——[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] `def` [illegible] `aa_bb` [illegible] `aaBb` [illegible]

[illegible] `self` [illegible]

[illegible]

`def __init__` [illegible] Python [illegible] java [illegible] `__new__` [illegible] init[illegible]

> [illegible] Python [illegible]

[illegible] name [illegible] name [illegible]

```
girl = Person('wangguniang')
```

girl [illegible] girl [illegible] name [illegible]

```
print girl.name
```

[illegible] `wangguniang`[illegible]

[illegible]“[illegible]”[illegible] girl [illegible] name [illegible]"wangguniang"[illegible]

[illegible]

```
def   init__(self,*args):
    pass
```

[illegible]*args [illegible]self [illegible]

[illegible]

```
 class Person:
    def   init  (self, name, lang="golang", 
website="www.google.com"):
        self.name = name
        self.lang = lang
        self.website = website
        self.email = "qiwsir@gmail.com"

laoqi = Person("LaoQi")
info = 
Person("qiwsir",lang="python",website="qiwsir.gi
thub.io")

print "laoqi.name=",laoqi.name
print "info.name=",info.name
print "-------"
print "laoqi.lang=",laoqi.lang
print "info.lang=",info.lang
print "-------"
print "laoqi.website=",laoqi.website
print "info.website=",info.website

#□□□□

laoqi.name= LaoQi
info.name= qiwsir
-------
laoqi.lang= golang
info.lang= python
-------
laoqi.website= www.google.com
info.website= qiwsir.github.io
```

□□□□□□□□□□□□□“□□□□□□”□□“□□”□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□

[illegible]

[illegible]`def getName(self)`,`def color(self, color)`[illegible] self [illegible]

```
def getName(self):
    return self.name
```

[illegible]

```
girl = Person('wangguniang')
name = girl.getName()
```

`girl.getName()`[illegible] girl [illegible] `getName(self)` [illegible] self [illegible] name [illegible] `name = "wangguniang"`[illegible]

[illegible]

```
def color(self, color):
    print "%s is %s" % (self.name, color)
```

[illegible]

```
girl.color("white")
```

[illegible] self [illegible] color[illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

```

```
__metaclass__ = type              #[illegible]

class Person:                     #[illegible]
    def   init  (self, name):    #[illegible]
        self.name = name

    def getName(self):            #[illegible]
        return self.name

    def color(self, color):
        print "%s is %s" % (self.name, color)

girl = Person('wangguniang')       #[illegible]
name = girl.getName()             #[illegible]
print "the person's name is: ", name
girl.color("white")               #[illegible]

print "------"
print girl.name                   #[illegible]
```

[illegible]

```
 $ python 20701.py
the person's name is:  wangguniang
wangguniang is white
------
wangguniang
```

[illegible]

[illegible]

- “[illegible]”[illegible] Learning Python[illegible]

- [illegible]
- [illegible]`class Person`[illegible]class [illegible] Person[illegible]

[illegible]

self [illegible]

[illegible] self[illegible] self [illegible]

self [illegible]

[illegible] Person [illegible] `girl = Person("wangguniang")` [illegible]"wangguniang"[illegible]`__init__()` [illegible] Person [illegible] girl [illegible]:`object.attribute`[illegible] `girl.name` [illegible]`"wangguniang"`[illegible]“.”[illegible]

[illegible]"wangguniang"[illegible]

[illegible] self[illegible]

[illegible] self[illegible]——[illegible]self [illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8
```

```
__metaclass__ = type

class Person:
    def   init  (self, name):
        self.name = name
        print self           #□□
        print type(self)     #□□
```

□□

```
<  main  .Person object at 0xb7282cec>
<class '__main__.Person'>
```

□□□□□□self □□□□□□□□□□□□□□□□□□□□

self □□□□□□□□□□□ girl □□

```
self.name = name
```

□□□□□ self □□□□□□□□□□□□□□□□□□ name□□□□□□□□□□□□□□□□□ name □□□□□□□□□□`self.name` □□ name □□□□□□□□□ `name` □□ `self.xxxooo = name` □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□ girl □□□ self□girl □□ self □□□□□□□□□□□□

□□□self □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
#!/usr/bin/env Python
#coding:utf-8

__metaclass__ = type
```

```
class Person:
    def   init  (self, name):
        self.name = name
        self.email = "qiwsir@gmail.com"     #[illegible]
[illegible]

info = Person("qiwsir")              #[illegible]
[illegible]
print "info.name=",info.name
print "info.email=",info.email      #info [illegible]
self [illegible]
```

[illegible]

```
info.name= qiwsir
info.email= qiwsir@gmail.com    #[illegible]
```

[illegible] self [illegible] `self.attribute` [illegible] self [illegible] info [illegible] info.email [illegible] self [illegible]“[illegible]”[illegible] info [illegible] self [illegible]self [illegible]info [illegible]

[illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

[illegible](3)

[illegible]

[illegible]

[illegible]

```
>>> class A(object):
...     x = 7
...
```

[illegible]`(object)`[illegible] `x = 7`[illegible]

```
>>> A.x
7
```

[illegible] A [illegible] x [illegible] x [illegible] A [illegible]
[illegible]“[illegible]”[illegible]——[illegible]

```
>>> foo = A()
>>> foo.x
7
```

[illegible]“[illegible]”[illegible]
[illegible]

```
>>> foo.x += 1
>>> foo.x
```

```
8
>>> A.x
7
```

□□“□□□□□□□□□□□”□□□□`foo.x += 1` □□□□□□□□□□□□□□□ foo □□□□□□□□□□□□□□□□□□□□ foo.x□□□□□□□□□□□□ foo.x□□□□□□□□□□ foo.x □□“□□□”□□□□□□□□ foo.x□□□□□ 8.

```
>>> foo.x
8
>>> del foo.x
>>> foo.x
7
```

□□□□ foo.x“□□”□□□ foo.x□□□□□□□□□□□□□□□□□□□□□□□□□ foo.x □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> foo.y = foo.x + 1
>>> foo.y
8
>>> foo.x
7
```

foo.y □□□□□□□□□□□□□□□□□□□□□□□ foo.x□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> A.x += 1
>>> A.x
8
>>> foo.x
8
```

□□□□□□□□□□□□□□□□

[illegible]

```
>>> class B(object):
...     y = [1,2,3]
...
```

[illegible]

```
>>> B.y         #[illegible]
[1, 2, 3]
>>> bar = B()
>>> bar.y       #[illegible]
[1, 2, 3]

>>> bar.y.append(4)
>>> bar.y
[1, 2, 3, 4]
>>> B.y
[1, 2, 3, 4]

>>> B.y.append("aa")
>>> B.y
[1, 2, 3, 4, 'aa']
>>> bar.y
[1, 2, 3, 4, 'aa']
```

[illegible]

[illegible]

```
>>> foo = A()
>>> dir(foo)
['  class  ', '  delattr  ', '  dict  ',
'  doc  ', '  format  ', '  getattribute  ',
'  hash  ', '  init  ', '  module  ', '  new__',
'__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__sizeof__', '__str__',
'__subclasshook__', '__weakref__', 'x']
```

[illegible] A[illegible]x[illegible] `dir(A)` [illegible]

```
>>> A.y = "hello"
>>> foo.y
'hello'
```

[illegible]

```
>>> foo.z = "python"
>>> foo.z
'python'
>>> A.z
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: type object 'A' has no attribute
'z'
```

[illegible] `object.attribute` [illegible]

[illegible]

[illegible]

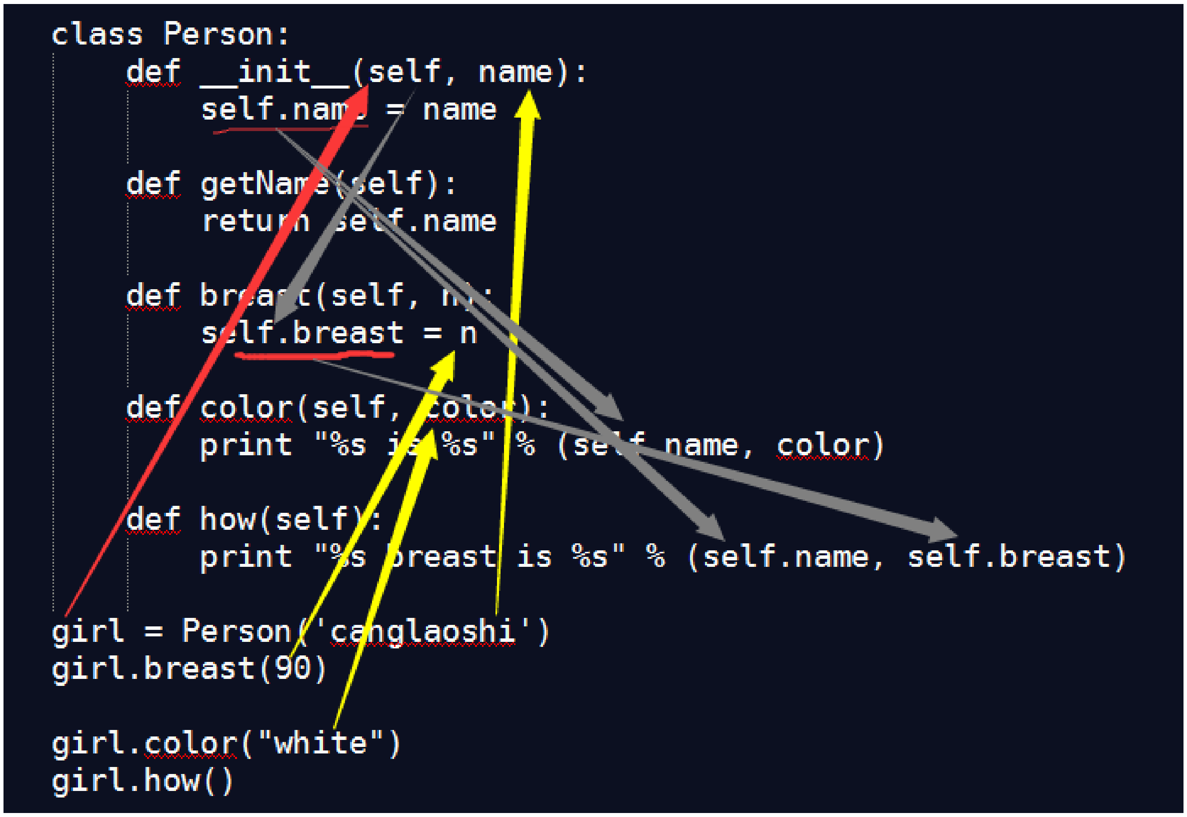
[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class Person:
    def __init__(self, name):
        self.name = name

    def getName(self):
        return self.name

    def breast(self, n):
        self.breast = n

    def color(self, color):
        print "%s is %s" % (self.name, color)

    def how(self):
        print "%s breast is %s" % (self.name,
self.breast)

girl = Person('wangguniang')
girl.breast(90)

girl.color("white")
girl.how()
```

[illegible]

```
$ python 20701.py
wangguniang is white
wangguniang breast is 90
```

[illegible]



```
class Person:
    def __init__(self, name):
        self.name = name

    def getName(self):
        return self.name

    def breast(self, n):
        self.breast = n

    def color(self, color):
        print "%s is %s" % (self.name, color)

    def how(self):
        print "%s breast is %s" % (self.name, self.breast)

girl = Person('canglaoshi')
girl.breast(90)

girl.color("white")
girl.how()
```

□□□□ `girl = Person('wangguniang')` □□□□□□□□□□□□□□girl □□□□□ Person □□□ self □□□□□□□□□□□□□□□“□□□□□ self □□□□□□ □□□□self □□”□□□□□"wangguniang"□□□□□□□□□□□□□□□□□□□ name □□□□□ self.name□□□□□□□□self □□□□□□□□□□□□□□□□□ `self.name` □□□□□□□□□□□□□□□□□□□□□□□□□ name □□□□□□ □"wangguniang"□

□□ Person □□□□□□□□□□ self □□□□□□□□□□□□□□□□□□ Python □□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ self □□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□ `self.name` □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ self □□□□□□□ `self.breast` □□□□□□ self □□□ □□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□ `return` □□□□□□□□□□□□□ `getName` □□□

□□□□□□(girl)□ self □□□□□□□□□□□□□□□□□□□ girl □□ self□□□□□□□ □□□

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class Person:
    def   init  (self, name):
        self.name = name

    def getName(self):
        #return self.name
        return girl.name    #□□□□□□□□□□□□□□□□□□□
□□□

girl = Person('wangguniang')
name = girl.getName()
print name
```

□□□□□□□□

```
wangguniang
```

□□□□□□□□□□□□□□□□□□ girl □□□□□□□□ self □□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□namespaces□□□□□□□□□□□□□□□□□□□□□□□□□□□□(2)□□“□□□□□”——□□□□□□□□□ class □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□ Python □□□□□□□□□□ `import this` □□□□□:

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the
rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation
to guess.
There should be one-- and preferably only one --
obvious way to do it.
Although that way may not be obvious at first
unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a
bad idea.
If the implementation is easy to explain, it may
be a good idea.
Namespaces are one honking great idea -- let's
do more of those!
```

□□□□□□□□□□□□Python □□□□□□□□□□□ Namespaces are one honking great idea -- let's do more of those!

□□□□□□□□□ Namespaces□□□□□□□□□□

[illegible][[illegible](2)[illegible] href="https://github.com/qiwsir/StarterLearningPython/blob/master/202.md")[illegible]

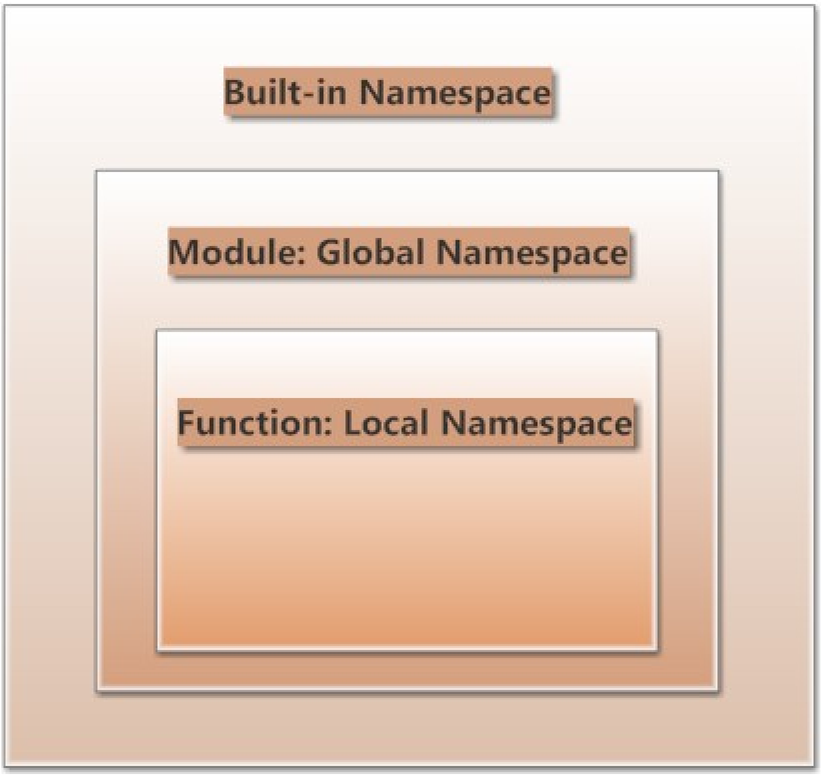
[illegible]

[illegible]

[illegible]

- [illegible](Built-in Namespaces)[illegible]Python [illegible] id(),[illegible]
- [illegible](Module:Global Namespaces)[illegible]
- [illegible](Function&Class: Local Namespaces)[illegible]

[illegible]


Built-in Namespace
Module: Global Namespace
Function: Local Namespace

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Local Namespaces --> Global Namesspaces --> Built-in Namesspaces

```
>>> def foo(num,str):
...     name = "qiwsir"
...     print locals()
...
>>> foo(221,"qiwsir.github.io")
{'num': 221, 'name': 'qiwsir', 'str': 'qiwsir.github.io'}
>>>
```

□□□□□□□□□□□□□□□□□□ `print locals()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ dictionary □□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `print globals()`□

□□□

[illegible] Python [illegible]“[illegible]”[illegible]

[illegible]

```
def outer_foo():
    b = 20
    def inner_foo():
        c = 30
a = 10
```

[illegible] inner_foo() [illegible] c [illegible] b [illegible] a [illegible] inner_foo() [illegible]b=50[illegible] b=20 [illegible]

```
#!/usr/bin/env Python
#coding:utf-8

def outer_foo():
    a = 10
    def inner_foo():
        a = 20
        print "inner_foo,a=",a      #a=20

    inner foo()
    print "outer_foo,a=",a          #a=10

a = 30
outer foo()
print "a=",a                #a=30

#[illegible]

inner_foo,a= 20
```

```
outer_foo,a= 10
a= 30
```

[illegible] global [illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible](4)

[illegible]——[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

> [illegible]Inheritance[illegible] [illegible] A“[illegible]”[illegible] B[illegible] A [illegible]“B [illegible]”[illegible] B [illegible]“A [illegible]”[illegible]“B [illegible] A [illegible]”[illegible]
>
> [illegible] [illegible]

[illegible]

- [illegible]
- [illegible]

[illegible]

> [illegible]OOP [illegible]

> [illegible] Python [illegible] Duck Type[illegible]
>
> [illegible]

[illegible]

[illegible]

[illegible] Python [illegible] object [illegible]

```
class NewStyle(object):
    pass
```

[illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class Person:
    def speak(self):
        print "I love you."

    def setHeight(self, n):
        self.length = n

    def breast(self, n):
        print "My breast is: ",n
```

```
class Girl(Person):
    def setHeight(self):
        print "The height is:1.70m ."

if   name   == " _main__":
    cang = Girl()
    cang.setHeight()
    cang.speak()
    cang.breast(90)
```

□□□□□□□□□□□□□□□□□□

```
 $ python 20901.py
The height is:1.70m .
I love you.
My breast is:  90
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□ Person□□□

□□□□□□□□□□□ Girl□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Girl □□□ Person□Girl □ Person □□□□□Person □ Girl □□□□□

□□□□□□□ Person□□□ Girl □□□□□□□ Person □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Girl □□□□□□□ Person □□□□□□□□□□□□□□ Person □□□□□□□□□□□□□□□□□□□ Girl □□□□□□□□□□□□□□□□□

□□□□□ Girl □□□□□□□□□□□ `cang.setHeight()`□□□□□□ Girl □□□□□ setHeight □□□□□□ Person □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Girl □□□□□□

□□□□□ Girl □□□□□□ speak □□□□□□□□□□□□□□ Person□□□ `cang.speak()` □□□□□ Person □□□□□□□□ `cang.breast(90)`□□□□□□□□□□ Girl □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

[illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class Person:
    def eye(self):
        print "two eyes"

    def breast(self, n):
        print "The breast is: ",n

class Girl:
    age = 28
    def color(self):
        print "The girl is white"

class HotGirl(Person, Girl):
    pass

if __name__ == "__main__":
    kong = HotGirl()
    kong.eye()
    kong.breast(90)
    kong.color()
    print kong.age
```

[illegible]Person [illegible] Girl[illegible] HotGirl [illegible]

□□□□□□ HotGirl□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□

```
$ python 20902.py
two eyes
The breast is:  90
The girl is white
28
```

□□□□□□□□□□□□ Girl □□□□□ `age = 28`□□□ HotGirl □□□□□□□□□□
□□□□□□□□□□□□□□ HotGirl □□□□□□□□□□ `kong.age` □□□□□□□□
□□

□□
□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□
□□
□□□

```
#!/usr/bin/env Python
# coding=utf-8

class K1(object):
    def foo(self):
        print "K1-foo"

class K2(object):
    def foo(self):
        print "K2-foo"
    def bar(self):
        print "K2-bar"
```

```
class J1(K1, K2):
    pass

class J2(K1, K2):
    def bar(self):
        print "J2-bar"

class C(J1, J2):
    pass

if __name__ == "__main__":
    print C.__mro__
    m = C()
    m.foo()
    m.bar()
```

[illegible]

```
 $ python 20904.py
(<class '__main__.C'>, <class '__main__.J1'>,
<class '__main__.J2'>, <class '__main__.K1'>,
<class '__main__.K2'>, <type 'object'>)
K1-foo
J2-bar
```

[illegible] `print C.__mro__` [illegible] foo() [illegible] J1[illegible] J2[illegible] J1 [illegible] K1[illegible] C==>J1==>J2==>K1[illegible]bar() [illegible] J2 [illegible]

[illegible]“[illegible]”[illegible]

[illegible] Python3.x [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

super [illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class Person:
    def   init  (self):
        self.height = 160

    def about(self, name):
        print "{} is about {}".format(name,
self.height)

class Girl(Person):
    def   init  (self):
        self.breast = 90

    def about(self, name):
        print "{} is a hot girl, she is about
{}, and her breast is {}".format(name,
self.height, self.breast)

if   name   == " _main__":
    cang = Girl()
    cang.about("wangguniang")
```

[illegible] Girl [illegible] Person[illegible] Girl [illegible] `self.breast = 90` [illegible] Person[illegible]Person [illegible] `self.height = 160` [illegible] Girl [illegible] about [illegible] `self.height` [illegible]

[illegible] Girl[illegible] `cang.about("wangguniang")` [illegible] `wangguniang is a hot girl, she is about 160, and her bereast is 90` [illegible]

```
$ python 20903.py
Traceback (most recent call last):
  File "20903.py", line 22, in <module>
    cang.about("wangguniang")
  File "20903.py", line 18, in about
    print "{} is a hot girl, she is about {}, and her breast is {}".format(name, self.height, self.breast)
AttributeError: 'Girl' object has no attribute 'height'
```

[illegible]

[illegible]

[illegible] `self.height` [illegible] Girl [illegible] Person [illegible]

[illegible] Girl[illegible] about [illegible], `__init__` [illegible] Person [illegible] `__init__` [illegible]

[illegible]

Python [illegible]super [illegible]

```
#!/usr/bin/env python
# coding=utf-8

__metaclass__ = type

class Person:
    def   init  (self):
        self.height = 160

    def about(self, name):
        print "{} is about {}".format(name,
self.height)

class Girl(Person):
    def   init  (self):
        super(Girl, self).__init__()
        self.breast = 90

    def about(self, name):
        print "{} is a hot girl, she is about
{}, and her breast is {}".format(name,
self.height, self.breast)
        super(Girl, self).about(name)

if   name   == " _main__":
    cang = Girl()
    cang.about("wangguniang")
```

□□□□□`  init  `□□□□□□□□□□□□□□□□□□`super(Girl, self).__init__()`□□□□super □□□□□□□□□□□□□□□□□□□□□□□self□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ about □□□□□□□□□□□□ about □□□

□□□□□

```
 $ python 20903.py
wangguniang is a hot girl, she is about 160, and
her breast is 90
wangguniang is about 160
```

□□□□□□□□super □□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□

[illegible]

[illegible]qiwsir@126.com,[illegible]

[illegible](5)

[illegible]

- [illegible] self[illegible]
- [illegible] self [illegible]

[illegible]

[illegible]

[illegible]

```
class Person(object):
    def foo(self):
        pass
```

[illegible] Person.foo() [illegible]

```
pp = Person()    #[illegible]
pp.foo()
```

[illegible] `pp.foo()` [illegible]

[illegible]

□□□(4)□□□□□□□□□□ super□□□□□□□□□□□□□□□□

```
#!/usr/bin/env python
# coding=utf-8

__metaclass__ = type

class Person:
    def   init  (self):
        self.height = 160

    def about(self, name):
        print "{} is about {}".format(name,
self.height)

class Girl(Person):
    def   init  (self):
        super(Girl, self).__init__()
        self.breast = 90

    def about(self, name):
        print "{} is a hot girl, she is about
{}, and her breast is {}".format(name,
self.height, self.breast)
        super(Girl, self).about(name)

if   name   == " _main__":
    cang = Girl()
    cang.about("wangguniang")
```

□□□ Girl □□□□□□□□□□ `init` □□□□□□□□□□□□□□□□□□□□□□□ `super(Girl, self).__init__()` □□□□□□□□□□□□□□□□□□□□□□□□□□

□□ super □□□□

[illegible]

[illegible] Girl [illegible] Person [illegible]

```
Person.__init__(self)
```

[illegible] Person [illegible] `__init__(self)` [illegible] super [illegible] super [illegible]

[illegible]

[illegible] self[illegible] super [illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class StaticMethod:
    @staticmethod
    def foo():
        print "This is static method foo()."

class ClassMethod:
    @classmethod
    def bar(cls):
        print "This is class method bar()."
        print "bar() is part of class:",
cls.__name__
```

```
if   name   == "  main  ":
    static foo = StaticMethod()    #[illegible]
    static foo.foo()               #[illegible]
    StaticMethod.foo()             #[illegible]
    print "********"
    class bar = ClassMethod()
    class bar.bar()
    ClassMethod.bar()
```

[illegible]“@”[illegible] Python [illegible]

- `@staticmethod` [illegible]
- `@classmethod` [illegible]

[illegible]

[illegible] def [illegible] self[illegible]“[illegible]”[illegible] self[illegible] self [illegible]

[illegible] self[illegible] cls [illegible]

[illegible] `StaticMethod.foo()` [illegible]

[illegible]

```
$ python 21001.py
This is static method foo().
This is static method foo().
********
This is class method bar().
```

```
bar() is part of class: ClassMethod
This is class method bar().
bar() is part of class: ClassMethod
```

[illegible]

[illegible]Python [illegible] staticmethod [illegible] classmethod [illegible]www.pythoncentral.io/difference-between-staticmethod-and-classmethod-in-Python/[illegible]
http://www.wklken.me/posts/2013/12/22/difference-between-staticmethod-and-classmethod-in-Python.html

[illegible]

Class vs static methods in Python

[illegible] staticmethod/classmethod,[illegible].

staticmethod [illegible] classmethod [illegible]"staticmethod"[illegible]"classmethod"

[illegible] Python [illegible]

Simple, static and class methods

[illegible] [illegible](instance methods), [illegible]

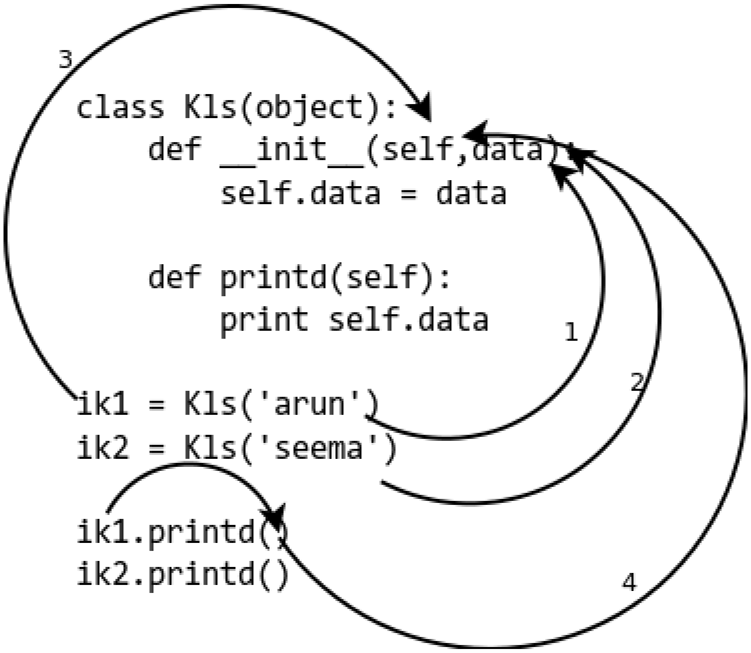
[illegible]

```
class Kls(object):
    def   init  (self, data):
        self.data = data

    def printd(self):
        print(self.data)

ik1 = Kls('arun')
ik2 = Kls('seema')

ik1.printd()
ik2.printd()
```

□□□□□:

```
arun
seema
```

□□□□□:



```
class Kls(object):
    def __init__(self,data):
        self.data = data

    def printd(self):
        print self.data

ik1 = Kls('arun')
ik2 = Kls('seema')

ik1.printd()
ik2.printd()
```

□□□□□□□:

> 1/2 [illegible]
>
> 3 self [illegible]
>
> 4 [illegible]

[illegible]

[illegible]

```
def get_no_of_instances(cls_obj):
    return cls_obj.no_inst

class Kls(object):
    no_inst = 0

    def __init__(self):
        Kls.no_inst = Kls.no_inst + 1

ik1 = Kls()
ik2 = Kls()

print(get_no_of_instances(Kls))
```

[illegible]

```
2
```

The Python @classmethod

[illegible].

[illegible],[illegible]:

```
 def iget no of instance(ins obj):
    return ins_obj.__class__.no_inst

class Kls(object):
    no_inst = 0

    def   init  (self):
        Kls.no_inst = Kls.no_inst + 1

ik1 = Kls()
ik2 = Kls()
print iget_no_of_instance(ik1)
```

□□

```
 2
```

□□□□ Python2.2 □□□□□□□□□ @classmethod □□□□□□□□□□□

```
 class Kls(object):
    no_inst = 0

    def   init  (self):
        Kls.no_inst = Kls.no_inst + 1

    @classmethod
    def get no of instance(cls_obj):
        return cls_obj.no_inst

ik1 = Kls()
ik2 = Kls()

print ik1.get no of instance()
print Kls.get_no_of_instance()
```

We get the following output:

```
2
2
```

The Python @staticmethod

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□.

□□□□□□□□□□□□□□□□□□□□.□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□:

```
IND = 'ON'

def checkind():
    return (IND == 'ON')

class Kls(object):
    def __init__(self,data):
        self.data = data

    def do_reset(self):
        if checkind():
            print('Reset done for:', self.data)

    def set_db(self):
        if checkind():
            self.db = 'new db connection'
            print('DB connection made for:',self.data)

ik1 = Kls(12)
ik1.do_reset()
ik1.set_db()
```

□□:

```
 Reset done for: 12
DB connection made for: 12
```

□□□□□□ @staticmethod, □□□□□□□□□□□□□

```
 IND = 'ON'

class Kls(object):
    def   init  (self, data):
        self.data = data

    @staticmethod
    def checkind():
        return (IND == 'ON')

    def do reset(self):
        if self.checkind():
            print('Reset done for:', self.data)

    def set db(self):
        if self.checkind():
            self.db = 'New db connection'
        print('DB connection made for: ',
self.data)

ik1 = Kls(12)
ik1.do reset()
ik1.set_db()
```

□□□□□:

```
 Reset done for: 12
DB connection made for: 12
```

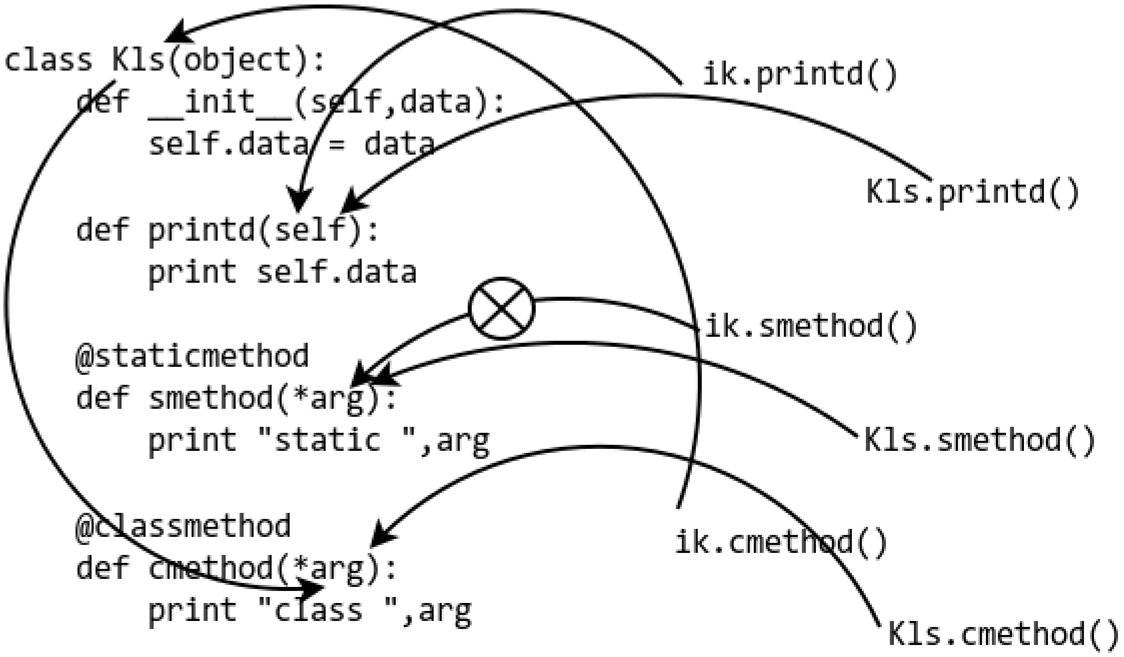
How @staticmethod and @classmethod are different

```
class Kls(object):
    def __init__(self, data):
        self.data = data

    def printd(self):
        print(self.data)

    @staticmethod
    def smethod(*arg):
        print('Static:', arg)

    @classmethod
    def cmethod(*arg):
        print('Class:', arg)
```

□□

```
>>> ik = Kls(23)
>>> ik.printd()
23
>>> ik.smethod()
Static: ()
>>> ik.cmethod()
Class: (<class '__main__.Kls'>,)
>>> Kls.printd()
TypeError: unbound method printd() must be
called with Kls instance as first argument (got
nothing instead)
>>> Kls.smethod()
Static: ()
>>> Kls.cmethod()
Class: (<class '__main__.Kls'>,)
```

[illegible]



```
class Kls(object):
    def __init__(self,data):
        self.data = data

    def printd(self):
        print self.data

    @staticmethod
    def smethod(*arg):
        print "static ",arg

    @classmethod
    def cmethod(*arg):
        print "class ",arg
```

ik.printd()
Kls.printd()
ik.smethod()
Kls.smethod()
ik.cmethod()
Kls.cmethod()

[illegible]

[illegible]

[illegible] help() [illegible]

```
 """This is python lesson"""

def start func(arg):
    """This is a function."""
    pass

class MyClass:
    """Thi is my class."""
    def my_method(self,arg):
        """This is my method."""
        pass
```

[illegible]

[illegible]“#”[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible]

[illegible]“[illegible]”[illegible] Python [illegible]

[illegible]

[illegible] Python [illegible]

[illegible] Python [illegible]

```
>>> "This is a book".count("s")
2
>>> [1,2,4,3,5,3].count(3)
2
```

[illegible] `count()` [illegible] count [illegible]

```
>>> f = lambda x,y:x+y
```

[illegible] lambda [illegible][[illegible](4)href="https://github.com/qiwsir/StarterLearningPython/blob/master/204.md")[illegible]

```
>>> f(2,3)
5
>>> f("qiw","sir")
```

```
'qiwsir'
>>> f(["python","java"],["c++","lisp"])
['python', 'java', 'c++', 'lisp']
```

[illegible] lambda [illegible] Pythonic [illegible] `+` [illegible]

[illegible]“[illegible]”[illegible]Python [illegible]

```
>>> f("qiw", 2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 1, in <lambda>
TypeError: cannot concatenate 'str' and 'int'
objects
```

[illegible]“[illegible]”[illegible]

“[illegible]”[illegible]:Polymorphism[illegible]“[illegible]”[illegible]

> [illegible]Polymorphism[illegible]

[illegible]“[illegible]”[illegible] Python [illegible] Pthonic [illegible]

[illegible] `repr()` [illegible]

```
>>> repr([1,2,3])
'[1, 2, 3]'
>>> repr(1)
'1'
```

```
>>> repr({"lang":"python"})
"{'lang': 'Python'}"
```

[illegible]

```
>>> def length(x):
...     print "The length of", repr(x), "is", len(x)
...

>>> length("how are you")
The length of 'how are you' is 11
>>> length([1,2,3])
The length of [1, 2, 3] is 3
>>> length({"lang":"python","book":"itdiffer.com"})
The length of {'lang': 'python', 'book': 'itdiffer.com'} is 2
```

[illegible]

```
>>> length(7)
The length of 7 is
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 2, in length
TypeError: object of type 'int' has no len()
```

[illegible] `object of type 'int' has no len()` [illegible]

[illegible] Python [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

```

```
"the code is from:
http://zetcode.com/lang/python/oop/"

__metaclass__ = type

class Animal:
    def   init  (self, name=""):
        self.name = name

    def talk(self):
        pass

class Cat(Animal):
    def talk(self):
        print "Meow!"

class Dog(Animal):
    def talk(self):
        print "Woof!"

a = Animal()
a.talk()

c = Cat("Missy")
c.talk()

d = Dog("Rocky")
d.talk()
```

[illegible]

```
 $ python 21101.py
Meow!
Woof!
```

[illegible] Cat [illegible] Dog [illegible] Animal[illegible] `talk()` [illegible]

[illegible]“[illegible]”(duck typeing)[illegible]

> [illegible]duck typing[illegible] James Whitcomb Riley [illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”

[illegible]

[illegible] type[illegible]isinstance [illegible] isubclass [illegible]

[illegible]

[illegible]

> [illegible]“[illegible]”[illegible]

[illegible]

“[illegible]”[illegible]“[illegible]”[illegible]

[illegible](Encapsulation)[illegible] object [illegible]

[illegible]“[illegible]”[illegible]

Python □□

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class ProtectMe:
    def __init__(self):
        self.me = "qiwsir"
        self.__name = "kivi"

    def __python(self):
        print "I love Python."

    def code(self):
        print "Which language do you like?"
        self.__python()

if __name__ == "__main__":
    p = ProtectMe()
    print p.me
    print p.__name
```

□□□□□□□□□□

```
$ python 21102.py
qiwsir
Traceback (most recent call last):
  File "21102.py", line 21, in <module>
    print p.__name
AttributeError: 'ProtectMe' object has no
attribute '__name'
```

□□□□□□□□□□□□□□□`__name` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□

```
if   name   == "  main__":
    p = ProtectMe()
    p.code()
    p.__python()
```

□□□□□□□□□□□ `p.code()` □□□□□□□□□□□□□□`"Which language do you like?"`□`"I love Python."`□`code()` □□□ `__python()` □□□□□□□□□□□□□□□□□□□□□□□ `p.__Python()` □□□□□
□□□□□□□□□□□□□□

```
$ python 21102.py
Which language do you like?
I love Python.
Traceback (most recent call last):
  File "21102.py", line 23, in <module>
    p.  python()
AttributeError: 'ProtectMe' object has no
attribute '__python'
```

□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□ `property` □□□

```
#!/usr/bin/env Python
# coding=utf-8

__metaclass__ = type

class ProtectMe:
    def   init  (self):
        self.me = "qiwsir"
        self.__name = "kivi"
    @property
    def name(self):
        return self.__name

if __name__ == "__main__":
    p = ProtectMe()
    print p.name
```

□□□□□

```
 $ python 21102.py
kivi
```

□□□□□□□□□□ `@property` □□□□□□□□□□□□□□□□□ `p.name` □□□□□□□□□□□□□□□□□□□□ `p.me` □□□□□□

□□□□□□□□□□“□□□□”□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

[illegible] (1)

[illegible]“From Beginner to Master”[illegible] Master [illegible] Master [illegible]“[illegible]”[illegible]

`__dict__`

[illegible](3)[illegible]“[illegible]”[illegible] `object.attribute` [illegible]

[illegible] Python [illegible]

```
>>> class A(object):
...     pass
...

>>> a = A()
>>> dir(a)
['__class__', '__delattr__', '__dict__',
'__doc__', '__format__', '__getattribute__',
'__hash__', '__init__', '__module__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__sizeof__', '__str__',
'__subclasshook__', '__weakref__']
```

```
>>> dir(A)
['  class  ', '  delattr  ', '  dict  ',
'  doc  ', '  format  ', '  getattribute  ',
'  hash  ', '  init  ', '  module  ', '  new__',
'  reduce  ', '  reduce ex  ', '  repr  ',
'  setattr  ', '  sizeof  ', '  str__',
'__subclasshook__', '__weakref__']
```

[illegible] `dir()` [illegible] `__dict__` [illegible]

```
>>> class Spring(object):
...     season = "the spring of class"
...

>>> Spring.  dict
dict proxy({'  dict__': <attribute '__dict__' of
'Spring' objects>,
'season': 'the spring of class',
'  module  ': '  main  ',
'  weakref  ': <attribute '__weakref__' of
'Spring' objects>,
'__doc__': None})
```

[illegible]

[illegible] Spring [illegible] `__dict__` [illegible] `'season'` [illegible]

```
>>> Spring.  dict  ['season']
'the spring of class'
>>> Spring.season
'the spring of class'
```

[illegible] `Spring.__dict__['season']` [illegible]

```
>>> s = Spring()
>>> s.__dict__
{}
```

[illegible]`__dict__`[illegible]

```
>>> s.season
'the spring of class'
```

[illegible]`Spring.season`[illegible]

```
>>> s.season = "the spring of instance"
>>> s.  dict
{'season': 'the spring of instance'}
```

[illegible]`s.season`[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

```
>>> s.  dict  ['season']
'the spring of instance'
>>> s.season
'the spring of instance'
```

[illegible]

```
>>> Spring.  dict  ['season']
'the spring of class'
>>> Spring.  dict
dict proxy({'  dict  ': <attribute '  dict  ' of
'Spring' objects>, 'season': 'the spring of
class', '  module  ': '  main  ', '  weakref  ':
<attribute '  weakref__' of 'Spring' objects>,
'  doc  ': None})
>>> Spring.season
'the spring of class'
```

Spring [illegible]

[illegible] `s.__dict__` [illegible]

```
>>> del s.season
>>> s.__dict__
{}
>>> s.season
'the spring of class'
```

[illegible]

[illegible] `__dict__` [illegible]

```
>>> s.lang = "python"
>>> s.  dict
{'lang': 'python'}
>>> s.  dict__['lang']
'python'
```

[illegible] `  dict__` [illegible] `Spring.  dict  ` [illegible] `Spring.lang` [illegible] `Spring.__dict__['lang']` [illegible]

```
>>> Spring.lang
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: type object 'Spring' has no
attribute 'lang'

>>> Spring.  dict  ['lang']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'lang'
```

[illegible]

```
>>> Spring.flower = "peach"
>>> Spring.  dict
dict proxy({'  module__': '__main__',
'flower': 'peach',
'season': 'the spring of class',
'  dict  ': <attribute '  dict  ' of 'Spring'
objects>, '  weakref  ': <attribute
'  weakref__' of 'Spring' objects>, '__doc__':
None})
>>> Spring.__dict__['flower']
'peach'
```

□□□`  dict__`□□□□□□□□□□□□□□'flower'□□□□□□□□□`__dict__`□□□□

```
>>> s.  dict
{'lang': 'python'}
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> s.flower
'peach'
```

□□□□□□□□□□□□□□□□□□□□□□□ `s.season` □□□□

□□□□□□□□□□□□□□□□□□□□`__dict__`□□

□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> class Spring(object):
...     def tree(self, x):
...         self.x = x
...         return self.x
...
>>> Spring.  dict
dict_proxy({'__dict__': <attribute '__dict__' of
'Spring' objects>,
'  weakref  ': <attribute '__weakref__' of
'Spring' objects>,
'  module  ': '  main  ',
'tree': <function tree at 0xb748fdf4>,
'__doc__': None})

>>> Spring.  dict  ['tree']
<function tree at 0xb748fdf4>
```

[illegible] `tree` [illegible]`__dict__`[illegible]

```
>>> t = Spring()
>>> t.__dict__
{}
```

[illegible]`__dict__`[illegible]

```
>>> t.tree("xiangzhangshu")
'xiangzhangshu'
```

[illegible](3)[illegible]“[illegible]”[illegible] `t` [illegible] `self` [illegible] `self.x = x`[illegible] x [illegible] self.x[illegible]

```
>>> t.  dict
{'x': 'xiangzhangshu'}
```

[illegible] `t` [illegible] `self` [illegible] (`t.tree('xiangzhangshu')`)[illegible](self[illegible])[illegible] t[illegible] self.x [illegible] `t.__dict__` [illegible]

[illegible]

```
>>> class Spring(object):
...     def tree(self, x):
```

```
...         return x
...
```

[illegible] x [illegible] self [illegible] return[illegible]

```
>>> s = Spring()
>>> s.tree("liushu")
'liushu'
>>> s.__dict__
{}
```

[illegible]

[illegible] Python [illegible]“[illegible]”[illegible] `object.attribute` [illegible]

[illegible]`__dict__`[illegible]

[illegible]`__dict__`[illegible]

[illegible]

__slots__

[illegible]`__slots__`[illegible]

```
>>> class Spring(object):
...     __slots__ = ("tree", "flower")
...
>>> dir(Spring)
['__class__', '__delattr__', '__doc__',
'  format  ', '  getattribute  ', '  hash__',
'  init  ', '  module  ', '  new  ',
'  reduce  ', '  reduce_ex  ', '  repr  ',
'  setattr  ', '  sizeof  ', '  slots  ',
'__str__', '__subclasshook__', 'flower', 'tree']
```

□□□□ `dir()` □□□□□□ `  dict__` □□□□□□□□□□□□□□□□□□ `__slots__` □ `__dict__` □□□□□□□□□□□□□□

```
>>> Spring.  slots__
('tree', 'flower')
```

□□□□□□□□ Spring □□□□□□□□□

```
>>> t = Spring()
>>> t.  slots
('tree', 'flower')
```

□□□□□□□□□ `__slots__` □□□□□□□□□□□□□ `__dict__` □□□□□□

```
>>> Spring.tree = "liushu"
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> t.tree = "guangyulan"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'Spring' object attribute 'tree'
is read-only
```

□□□□□□□□□□□□□□□□□□□□□ `tree` □□□□□□□□□□□□□□□

```
>>> t.tree
'liushu'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> Spring.tree = "guangyulan"
>>> t.tree
'guangyulan'
```

[illegible]

```
>>> t.flower = "haitanghua"
>>> t.flower
'haitanghua'
```

[illegible]

```
>>> Spring.flower
<member 'flower' of 'Spring' objects>
```

[illegible]

```
>>> Spring.flower = "ziteng"
>>> t.flower
'ziteng'
```

[illegible] `t.flower` [illegible]

```
>>> t.water = "green"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'Spring' object has no attribute
'water'
```

[illegible]

[illegible] `__slots__` [illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

□□□□ (2)

□□□□□□□□□□□□ `dict` □□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ `__dict__` □□□□□□□□□□□□□□□□□□ □□□□□

```
>>> class A(object):
...     pass
...
>>> a = A()
>>> a.x
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'A' object has no attribute 'x'
```

`x` □□□□□□□ `a.x` □□□□□□□□□□□□□□□□“'A' object has no attribute 'x'”

□□□

□□□□□□□□□□□□□□□□

`__getattr__` □ `__setattr__` □□□□□□

□□□□□□□□□□ `a.x` □□□□□□□□□□□□□□□□□□□□□□“□□”□□□□“□□□□□”□□□□“□□□□”□□□“□□”□□□ Python □□□□□□□□□□“□□”□□□

- `__setattr__(self,name,value)` [illegible] name [illegible]
- `__getattr__(self,name)` [illegible] name [illegible]
- `__getattribute__(self,name)` [illegible] name [illegible] name [illegible]
- `__delattr__(self,name)` [illegible] name[illegible]

[illegible]

```
>>> class A(object):
...     def   getattr  (self, name):
...         print "You use getattr"
...     def   setattr  (self, name, value):
...         print "You use setattr"
...         self.__dict__[name] = value
...
```

[illegible] A [illegible]

```
>>> a = A()
>>> a.x
You use getattr
```

`a.x` [illegible] `__getattr__(self, name)` [illegible] `x` [illegible] `__dict__` [illegible] `__getattr__` [illegible]“[illegible]”[illegible]

```
>>> a.x = 7
You use setattr
```

[illegible] `  setattr  (self, name, value)` [illegible] `self.  dict__[name] = value` [illegible] `__dict__` [illegible]

```
>>> a.x
7
```

□□□□□□□□□`__dict__`□□□

□□□□□□□□□□□□□`__getattribute__(self, name)`□□□□□□□
□□□□□□□□□□□□□□□□□□□□

```
>>> class B(object):
...     def   getattribute  (self, name):
...         print "you are useing getattribute"
...         return object.__getattribute__(self,
name)
...
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `return object.  getattribute__(self, name)`□□□□□□ `return self.  dict  [name]`□□□□□□□□□□□□□□□□□□ `self.__dict__`□□□□□□□□□□□□□□□`**getattribute**``□□□□□□□
□□□□□□□□□□□□□□□□□

```
>>> b = B()
>>> b.y
you are useing getattribute
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 4, in   getattribute
AttributeError: 'B' object has no attribute 'y'
>>> b.two
you are useing getattribute
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 4, in   getattribute _
AttributeError: 'B' object has no attribute
'two'
```

□□□□□□□□□□□□□□□□□`__getattribute__`□□□□□□□□□□□□□□□

```
>>> b.y = 8
>>> b.y
you are useing getattribute
8
```

□□□□□□□□□□□□□`__dict__`□□□□□□□□□□□□□□□□□□□□□
`__dict__`□□□□□□□□□

□□□
□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

"""
study __getattr__ and __setattr__
"""

class Rectangle(object):
    """
    the width and length of Rectangle
    """
    def   init  (self):
        self.width = 0
        self.length = 0

    def setSize(self, size):
        self.width, self.length = size
    def getSize(self):
        return self.width, self.length

if   name   == "  main__":
    r = Rectangle()
    r.width = 3
    r.length = 4
    print r.getSize()
    r.setSize( (30, 40) )
    print r.width
    print r.length
```

□□□□□□□Beginning Python:From Novice to Professional,Second Edittion□□by Magnus Lie Hetland□□□□□□□□□□□□□□□□

```
$ python 21301.py
(3, 4)
30
40
```

□□

```
#!/usr/bin/env Python
# coding=utf-8

"""
study __getattr__ and __setattr__
"""

class Rectangle(object):
    """
    the width and length of Rectangle
    """
    def   init  (self):
        self.width = 0
        self.length = 0

    def setSize(self, size):
        self.width, self.length = size
    def getSize(self):
        return self.width, self.length
    size = property(getSize, setSize)

if   name   == "  main__":
    r = Rectangle()
    r.width = 3
    r.length = 4
    print r.size
    r.size = 30, 40
    print r.width
    print r.length
```

□□□□□□□□□□□□□□□□□□□□□ `size = property(getSize, setSize)` □□□□□□□□□□□□□□□□□□□ `r.getSize()` □□□□□ `r.size` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ property □□□□□□□□□□□□□□□□□□□□

□□□

```
#!/usr/bin/env Python
# coding=utf-8

class NewRectangle(object):
    def   init  (self):
        self.width = 0
        self.length = 0

    def   setattr  (self, name, value):
        if name == "size":
            self.width, self.length = value
        else:
            self.__dict__[name] = value

    def   getattr  (self, name):
        if name == "size":
            return self.width, self.length
        else:
            raise AttributeError

if __name__ == "__main__":
    r = NewRectangle()
    r.width = 3
    r.length = 4
    print r.size
    r.size = 30, 40
    print r.width
    print r.length
```

□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Python Attributes and Methods□□□□□□□□□ Python □□□□□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`__dict__`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

class A(object):
    author = "qiwsir"
    def __getattr__(self, name):
        if name != "author":
            return "from starter to master."

if __name__ == "__main__":
    a = A()
    print a.author
    print a.lang
```

□□□□□

```
$ python 21302.py
qiwsir
from starter to master.
```

□ `a = A()` □□□□□□□□□□□□□□□□□□□□□ `__dict__` □□□□□□□□□□□□□□□□□□□□□□□ `a.author` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `"qiwsir"` □□□□□□ `a.lang` □□□□□□□□□□□□□□□□□□□□□□□□□□□□ `__getattr__()` □□□□□□□□□□□□□□□□□□□□ `__getattr__()` □□□□□□□□□□□□□□□□□□□ AttributeError□□□□□□□□□□□

□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

[illegible]

[illegible]

[illegible] `iter()` [illegible] `next()` [illegible] for [illegible]

[illegible] `dir(list)`,`dir(tuple)`,`dir(file)`,`dir(dict)` [illegible] `__iter__` [illegible]iterator[illegible] iter() [illegible]

[illegible]`__iter__`[illegible](iterator potocol)[illegible] `next()` [illegible]

> [illegible] Python3.x[illegible]`__next__()` [illegible] `next()` [illegible] Python3.x [illegible] next()[illegible] `next(it)` [illegible] python2.x [illegible] `it.next()`[illegible]it [illegible]

[illegible] list[illegible]tuple[illegible]file[illegible]dict [illegible]`__iter__()`[illegible] Python [illegible]

[illegible] python [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

"""
the interator as range()
"""
```

```
class MyRange(object):
    def   init  (self, n):
        self.i = 0
        self.n = n

    def   iter  (self):
        return self

    def next(self):
        if self.i < self.n:
            i = self.i
            self.i += 1
            return i
        else:
            raise StopIteration()

if   name   == "  main__":
    x = MyRange(7)
    print "x.next()==>", x.next()
    print "x.next()==>", x.next()
    print "------for loop--------"
    for i in x:
        print i
```

[illegible]

```
 $ python 21401.py
x.next()==> 0
x.next()==> 1
------for loop--------
2
3
4
5
6
```

[illegible] `range()` [illegible]

[illegible] MyRange [illegible] `__init__()` [illegible] [illegible](2)[illegible] [illegible]

`__iter__()` [illegible] `__iter__()` [illegible]

[illegible] `next()` [illegible] `StopIteration()` [illegible]

[illegible]

```
if __name__ == "__main__":
    x = MyRange(7)
    print list(x)
    print "x.next()==>", x.next()
```

[illegible]

```
$ python 21401.py
[0, 1, 2, 3, 4, 5, 6]
x.next()==>
Traceback (most recent call last):
  File "21401.py", line 26, in <module>
    print "x.next()==>", x.next()
  File "21401.py", line 21, in next
    raise StopIteration()
StopIteration
```

[illegible] `print list(x)` [illegible] [illegible] [illegible] `next()` [illegible] `StopIteration()` [illegible]

[illegible]

[illegible] 4[illegible]:

```
#!/usr/bin/env Python
# coding=utf-8
"""
compute Fibonacci by iterator
"""
__metaclass__ = type

class Fibs:
    def   init  (self, max):
        self.max = max
        self.a = 0
        self.b = 1

    def   iter  (self):
        return self

    def next(self):
        fib = self.a
        if fib > self.max:
            raise StopIteration
        self.a, self.b = self.b, self.a + self.b
        return fib

if   name   == "  main__":
    fibs = Fibs(5)
    print list(fibs)
```

[illegible]

```
$ python 21402.py
[0, 1, 1, 2, 3, 5]
```

> □□□□□□□□□□□□□□□□□□□□□□ 1000 □□□□□□□□□□□□□□□□□□
> □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□`range()` □ `xrange()`□□
□□□□□□□□□□□□□□□

```
range(...)
    range(stop) -> list of integers
    range(start, stop[, step]) -> list of
integers

>>> dir(range)
['__call__', '__class__', '__cmp__',
'__delattr__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__',
'__gt__', '__hash__', '__init__', '__le__',
'__lt__', '__module__', '__name__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__',
'__repr__', '__self__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__']
```

□ `range()` □□□□□□□□□□□□□□□□□□□□□□□□□□
`help(xrange)` □□□

```
class xrange(object)
 |  xrange(stop) -> xrange object
 |  xrange(start, stop[, step]) -> xrange object
 |
 |  Like range(), but instead of returning a
list, returns an object that
 |  generates the numbers in the range on
demand.  For looping, this is
 |  slightly faster than range() and more memory
efficient.
```

`xrange()` [illegible] `range()` [illegible] `range()` [illegible]“slightly faster than range() and more memory efficient”[illegible]

```
>>> dir(xrange)
['  class  ', '  delattr  ', '  doc  ',
'  format  ', '  getattribute  ', '  getitem  ',
'  hash  ', '  init  ', '  iter  ', '  len__',
'  new  ', '  reduce  ', '  reduce ex  ',
'  repr  ', '  reversed  ', '  setattr  ',
'__sizeof__', '__str__', '__subclasshook__']
```

[illegible]`__iter__`[illegible]

[illegible] `range()` [illegible] `xrange()` [illegible]

[illegible] `zip()` [illegible]

```
>>> a = ["name", "age"]
>>> b = ["qiwsir", 40]
>>> zip(a,b)
[('name', 'qiwsir'), ('age', 40)]
```

[illegible]

```
>>> zip(range(4), xrange(100000000))
[(0, 0), (1, 1), (2, 2), (3, 3)]
```

[illegible] `range(4)` [illegible] 4 [illegible] `range(100000000)` [illegible]

[illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

[illegible]

[illegible]generator[illegible] Python [illegible]——[illegible]——[illegible]——[illegible] Python [illegible]

[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

```
>>> my_generator = (x*x for x in range(4))
```

[illegible]

```
>>> my_list = [x*x for x in range(4)]
```

[illegible] `[]` [illegible] `()` [illegible]

```
>>> dir(my generator)
['  class  ', '  delattr  ', '  doc  ',
'  format  ', '__getattribute__', '__hash__',
'  init  ',
'  iter  ',
'  name  ', '  new  ', '  reduce  ',
'  reduce ex  ', '  repr  ', '  setattr  ',
'__sizeof__', '__str__', '__subclasshook__',
'close', 'gi_code', 'gi_frame', 'gi_running',
'next',
'send', 'throw']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`__inter__()`□ `next()` □□□□□□□□□□□□□□□□□□□□□ for □□□□□□□□□□

```
>>> for i in my_generator:
...     print i
...
0
1
4
9
>>> for i in my_generator:
...     print i
...
```

□□□□□□□□□□□ my_generator □□□

□□□□□□□□□□□□□□

```
>>> for i in my_list:
...     print i
...
0
1
4
9
>>> for i in my_list:
...     print i
...
0
1
```

```
4
9
```

[illegible] `[]` [illegible] `()` [illegible]“[illegible]”[illegible]

[illegible]

```
>>> sum(i*i for i in range(10))
285
```

[illegible] `sum()` [illegible]

```
>>> sum([i*i for i in range(10)])
285
```

[illegible]

[illegible]

yield [illegible]“[illegible]”[illegible] Python [illegible]

```
>>> def g():
...     yield 0
...     yield 1
...     yield 2
...
>>> g
<function g at 0xb71f3b8c>
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ yield □□□□□□□□□□□
□□

```
>>> ge = g()
>>> ge
<generator object g at 0xb7200edc>
>>> type(ge)
<type 'generator'>
```

□□□□□□□□□□□□□□□□(generator)□□□□□□

```
>>> dir(ge)
['__class__', '__delattr__', '__doc__',
'__format__', '__getattribute__', '__hash__',
'__init__', '__iter__', '__name__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__',
'__setattr__', '__sizeof__', '__str__',
'__subclasshook__', 'close', 'gi_code',
'gi_frame', 'gi_running', 'next', 'send',
'throw']
```

□□□□□□`__iter__()` □ `next()`□□□□□□□□□□□□□□□□□□□

```
>>> ge.next()
0
>>> ge.next()
1
>>> ge.next()
2
>>> ge.next()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
StopIteration
```

□□□□□□□□□□□□□□□□□ yield □□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□

□□□□□ yield □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`__inter__()` □ `next()`□□□□□□□ yield □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□

yield □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

1. `ge = g()`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
2. `ge.next()`□□□□□□□□□□□□□□□□□□□□□ yield □□□□□□□□ □□□□□□□□□□□□□□□
3. `ge.next()`□□□□□□□□□□□□□□□□□□□□□ yield □□□□□□□□□ □□□
4. `gen.next()`□□□□□□□□□
5. `gene.next()`□□□□□□□□□□□□□□□□□□□□□□□□□□ `next()` □□□□□

□□□□□□□□□□□□ yield □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ return □□□□□□□□□□□□

yield

□□□□□ yield □ return □□□□□□□□□□□□□□□□□□□

```
>>> def r return(n):
...     print "You taked me."
...     while n > 0:
...         print "before return"
...         return n
...         n -= 1
...         print "after return"
...
>>> rr = r_return(3)
```

```
You taked me.
before return
>>> rr
3
```

□□□□□□□□□□□□□□□□□□`rr = r_return(3)`□□□□□□□□□□□□□□□□□□ return□□□□□□□□□□□□□□□□□□□□□□□ return □□□□□□□□□□□□□□□□ return □□□□□□□□□□□□□□□□□□□□□□□(2)□□□□□□□□□□□□□

□□□ return □□ yield□

```
>>> def y_yield(n):
...     print "You taked me."
...     while n > 0:
...         print "before yield"
...         yield n
...         n -= 1
...         print "after yield"
...
>>> yy = y_yield(3)    #□□□□□□□□□□
>>> yy.next()          #□□□□
You taked me.
before yield
3                      #□□ yield□□□□□□□□
>>> yy.next()          #□□□□□□□□□□□□□
after yield
before yield
2                      #□□□ yield□□□□□□□□
>>> yy.next()          #□□□□□□
after yield
before yield
1
>>> yy.next()
after yield            #□□□□□□□□□□□□□
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
StopIteration
```

[illegible] yield [illegible] return [illegible]

[illegible] return[illegible] yield[illegible] return[illegible] SoptIteration [illegible]

[illegible] yield[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

def fibs(max):
    """
    [illegible]
    """
    n, a, b = 0, 0, 1
    while n < max:
        yield b
        a, b = b, a + b
        n = n + 1

if   name   == "__main__":
    f = fibs(10)
    for i in f:
        print i ,
```

[illegible]

```
$ python 21501.py
1 1 2 3 5 8 13 21 34 55
```

[illegible]

[illegible] yield [illegible]

[illegible]

[illegible] python2.5 [illegible]“[illegible]”[illegible]

```
>>> def repeater(n):
...     while True:
...         n = (yield n)
...
>>> r = repeater(4)
>>> r.next()
4
>>> r.send("hello")
'hello'
```

[illegible] `r.next()` [illegible] `yield n` [illegible] `n = (yield n)` [illegible] `yield n` [illegible] n[illegible]

[illegible] `r.send("hello")` [illegible] `n = (yield n)` [illegible] send() [illegible]

[illegible] `r.next()` [illegible]

```
>>> r.next()
```

[illegible] None[illegible] `r.next()` [illegible]yield [illegible] None.

□□□□□send() □□□□□□□□□□□□□□□□□□□□□□ yield □□□□□□□□□□□
□□□□

```
>>> s = repeater(5)
>>> s.send("how")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can't send non-None value to a just-
started generator
```

□□□□□□□□□□□□□□ None□

```
>>> s.send(None)
5
```

□□□□□□□□□□□□□□□□□

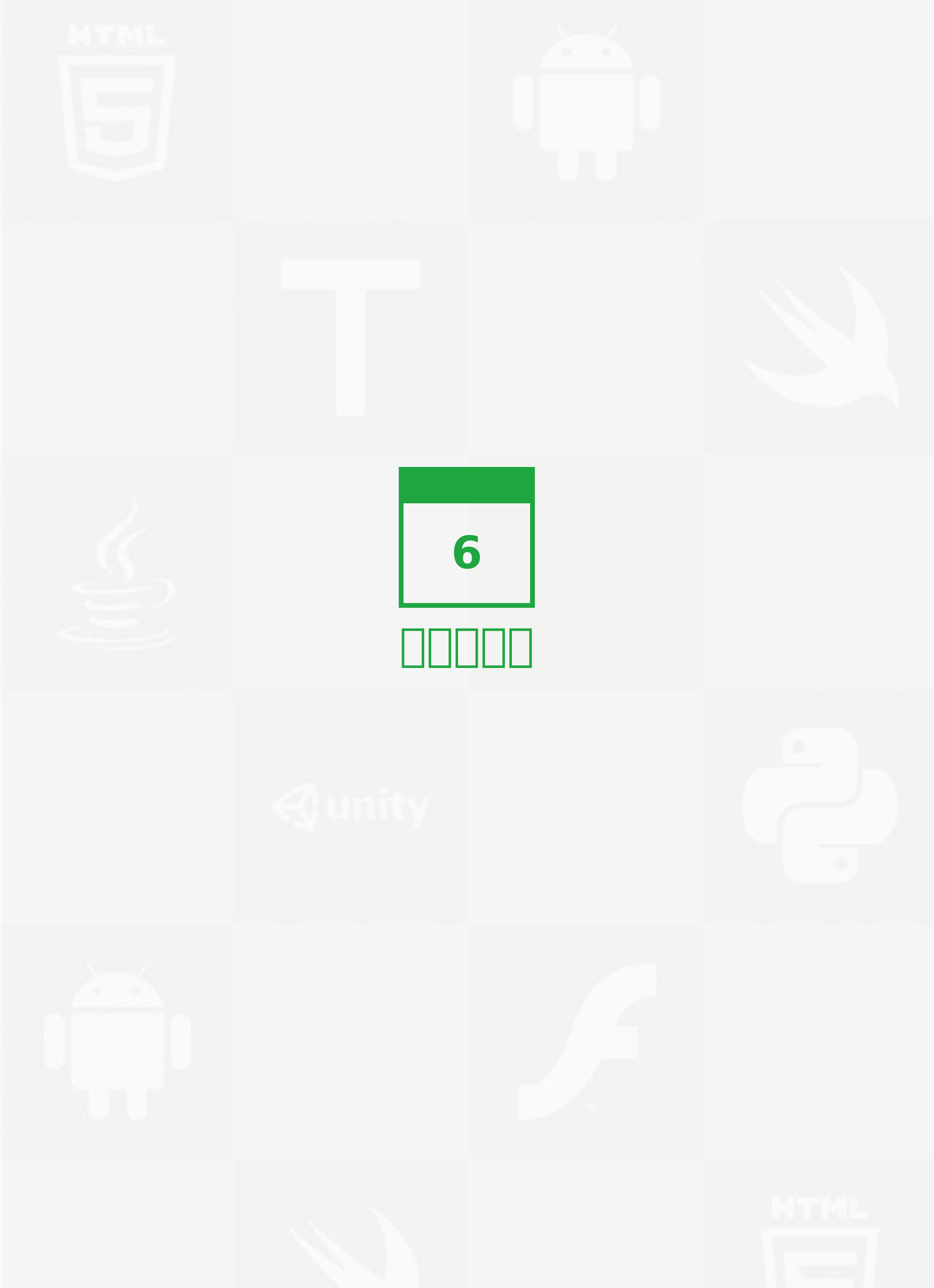
□□□□□□□□□□close() □ throw()

- throw(type, value=None, traceback=None):□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ yield □□□□□□
- close()□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□



6

[illegible]

□□□□□ (1)

□□

□□

Python □□□□□□□□□□□(syntax errors)□□□□

```
>>> for i in range(10)
  File "<stdin>", line 1
    for i in range(10)
                     ^
SyntaxError: invalid syntax
```

□□□□□□□□□□□`:`□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□`File "<stdin>", line 1`□□□□□□□□`^`□□□□□□□□□□□`:`□□□□□□□□□□□

□□□

□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□□□□ 0 □□□□□□□□□□□□□□□□□□□□□

```
>>> 1/0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: integer division or modulo by
zero
```

[illegible] Python [illegible]“[illegible](Traceback)”[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

[illegible]	[illegible]
NameError	[illegible]
ZeroDivisionError	[illegible] 0
SyntaxError	[illegible]
IndexError	[illegible]
KeyError	[illegible]
IOError	[illegible]
AttributeError	[illegible]

[illegible]

NameError

```
>>> bar
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'bar' is not defined
```

Python □□
□□□□□□□□□□□□□□□□□□□□□□□

ZeroDivisionError

```
>>> 1/0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: integer division or modulo by
zero
```

□□

SyntaxError

```
>>> for i in range(10)
  File "<stdin>", line 1
    for i in range(10)
                     ^
SyntaxError: invalid syntax
```

□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python
□□
□□□

IndexError

```
>>> a = [1,2,3]
>>> a[4]
```

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range

>>> d = {"python":"itdiffer.com"}
>>> d["java"]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'java'
```

□□□□□□“□□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□

IOError

```
>>> f = open("foo")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IOError: [Errno 2] No such file or directory:
'foo'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□ computer □□
□□□□□□□□Python □□□□□□□□□□□□□□□□□□

AttributeError

```
>>> class A(object): pass
...
>>> a = A()
>>> a.foo
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
```

```
AttributeError: 'A' object has no attribute
'foo'
```

[illegible]

[illegible]Python [illegible]

```
>>> range("aaa")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: range() integer end argument
expected, got str.
```

[illegible] Python [illegible] `dir()`[illegible]`help()` [illegible]google [illegible]

[illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

while 1:
    print "this is a division program."
    c = raw_input("input 'c' continue, otherwise
logout:")
    if c == 'c':
        a = raw_input("first number:")
        b = raw_input("second number:")
        try:
            print float(a)/float(b)
            print "*************************"
        except ZeroDivisionError:
            print "The second number can't be 
zero!"
            print "*************************"
    else:
        break
```

[illegible]

```
 $ python 21601.py
this is a division program.
input 'c' continue, otherwise logout:c
first number:5
second number:2
2.5
*************************
this is a division program.
input 'c' continue, otherwise logout:c
first number:5
second number:0
The second number can't be zero!
*************************
this is a division program.
input 'c' continue, otherwise logout:d
$
```

[illegible] 0 [illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible] `try...except...`[illegible]

[illegible] try [illegible] except [illegible]except [illegible] try [illegible] try [illegible] except [illegible]

except □□□□□□□□□□□□□□□□□□□□□□□□□□□□ try □□□□□□□□□□□
□□ except□

□ except □□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

class Calculator(object):
    is raise = False
    def calc(self, express):
        try:
            return eval(express)
        except ZeroDivisionError:
            if self.is raise:
                print "zero can not be
division."
            else:
                raise
```

□□□□□□□□□□□ `eval()` □□□□□□□

```
eval(...)
    eval(source[, globals[, locals]]) -> value

    Evaluate the source in the context of
globals and locals.
    The source may be a string representing a
Python expression
    or a code object as returned by compile().
    The globals must be a dictionary and locals
can be any mapping,
    defaulting to the current globals and
locals.
    If only globals is given, locals defaults to
it.
```

□□□

```
>>> eval("3+5")
8
```

□□□□ except □□□□□□□ `raise` □□□□□□□□□□□□□□□□□□□□□□□□□□□□except □□□□□□□□□□□□□□□□□□□□□□□□□□□

```
if   name   == "  main__":
    c = Calculator()
    print c.calc("8/0")
```

□□□ `is_raise = False` □□□□

```
$ python 21602.py
Traceback (most recent call last):
  File "21602.py", line 17, in <module>
    print c.calc("8/0")
  File "21602.py", line 8, in calc
    return eval(express)
  File "<string>", line 1, in <module>
ZeroDivisionError: integer division or modulo by
zero
```

□□□ `is_raise` □□□□ True□□□□□□□

```
if   name   == "  main__":
    c = Calculator()
    c.is raise = True    #□□□□□□□□
    print c.calc("8/0")
```

□□□□□

```
$ python 21602.py
zero can not be division.
None
```

[illegible] None [illegible] `c.calc("8/0")` [illegible] `print c.calc("8/0")` [illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

[illegible] (2)

try...except...[illegible]

[illegible]

[illegible] except [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

while 1:
    print "this is a division program."
    c = raw_input("input 'c' continue, otherwise logout:")
    if c == 'c':
        a = raw_input("first number:")
        b = raw_input("second number:")
        try:
            print float(a)/float(b)
            print "*************************"
        except ZeroDivisionError:
            print "The second number can't be zero!"
            print "*************************"
        except ValueError:
            print "please input number."
            print "************************"
    else:
        break
```

[illegible] except [illegible]

```
$ python 21701.py
this is a division program.
input 'c' continue, otherwise logout:c
first number:3
second number:"hello"        #[illegible]
please input number.         #[illegible]
************************
this is a division program.
input 'c' continue, otherwise logout:c
first number:4
second number:0
The second number can't be zero!
*************************
this is a division program.
input 'c' continue, otherwise logout:4
$
```

[illegible] except[illegible] try [illegible] except [illegible] except [illegible]“[illegible]”[illegible]

[illegible] except [illegible] except [illegible] except [illegible]

```
except (ZeroDivisionError, ValueError):
    print "please input rightly."
    print "********************"
```

[illegible]

```
 $ python 21701.py
this is a division program.
input 'c' continue, otherwise logout:c
first number:2
second number:0           #□□□□
please input rightly.
********************
this is a division program.
input 'c' continue, otherwise logout:c
first number:3
second number:a           #□□
please input rightly.
********************
this is a division program.
input 'c' continue, otherwise logout:d
$
```

□□□□□□□except □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□Python □□□□□□□□□□□□□□□□□

```
 while 1:
    print "this is a division program."
    c = raw_input("input 'c' continue, otherwise
logout:")
    if c == 'c':
        a = raw_input("first number:")
        b = raw_input("second number:")
        try:
            print float(a)/float(b)
            print "*************************"
        except (ZeroDivisionError, ValueError),
e:
            print e
            print "********************"
    else:
        break
```

□□□□□□□□□□□□

```
$ python 21702.py
this is a division program.
input 'c' continue, otherwise logout:c
first number:2
second number:a                              #□□
could not convert string to float: a
********************
this is a division program.
input 'c' continue, otherwise logout:c
first number:2
second number:0                              #□□
float division by zero
********************
this is a division program.
input 'c' continue, otherwise logout:d
$
```

> □ Python3.x □□□□□□□□`except (ZeroDivisionError, ValueError) as e:`

□□`execpt:`□□`except Exception, e`□□□□□□□□□□□□□

else □□

□□`try...except...`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ else □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> try:
...     print "I am try"
... except:
...     print "I am except"
... else:
...     print "I am else"
...
I am try
I am else
```

□□□□□□□□□□□□□ else □□□□□□□□□□□ try□□ except □□□□□□ else □□□□

```
>>> try:
...     print 1/0
... except:
...     print "I am except"
... else:
...     print "I am else"
...
I am except
```

□□□ else □□□□□□□

□□□ else □□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8
while 1:
    try:
        x = raw_input("the first number:")
        y = raw_input("the second number:")

        r = float(x)/float(y)
```

```
            print r
        except Exception, e:
            print e
            print "try again."
        else:
            break
```

□□□□□□□

```
 $ python 21703.py
the first number:2
the second number:0        #□□□□□ except
float division by zero
try again.                 #□□
the first number:2
the second number:a        #□□
could not convert string to float: a
try again.
the first number:4
the second number:2        #□□□□□ try
2.0                        #□□ else□break□□□□□
$
```

□□□□□□□□□□

□□□□□□□ except □□□□□□□□□□□□□□□□□□ `except Exception, e` □□□□□□□□□□□□□□□□□□□□□□□□ e□□□□ `print e` □□□□□□□□□□

finally

finally □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ finally□□□□□□□□ try□□□ except□□□□□□□□□□□□□□□ finally □□□□□□□□□□□□□□□□

```
>>> x = 10

>>> try:
...     x = 1/0
... except Exception, e:
...     print e
... finally:
...     print "del x"
...     del x
...
integer division or modulo by zero
del x
```

[illegible] x [illegible]

```
>>> x
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'x' is not defined
```

[illegible]

```
try:
    do something
except:
    do something
else:
    do something
finally
    do something
```

[illegible]

`try...except...`□□□□□□□□□□ `if...else..` □□

□□□

□□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

□□□□□ (3)

□□
□□□ from beginner to master□

assert

```
>>> assert 1==1
>>> assert 1==0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AssertionError
```

□□□□□□□□□□□□□□ assert □□□□

assert□□□□□□“□□”□□□assert □□□□□□□□□□□□□□□□□□□□□□□□□
□□

assert □□
□□

```
#!/usr/bin/env Python
# coding=utf-8

class Account(object):
    def   init  (self, number):
        self.number = number
        self.balance = 0
    def deposit(self, amount):
        assert amount > 0
        self.balance += balance

    def withdraw(self, amount):
        assert amount > 0
        if amount <= self.balance:
            self.balance -= amount
        else:
            print "balance is not enough."
```

□□□□□□□deposit() □ withdraw() □□□□□ amount □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
 if   name   == "  main__":
    a = Account(1000)
    a.deposit(-10)
```

□□□□□□□

```
 $ python 21801.py
Traceback (most recent call last):
  File "21801.py", line 22, in <module>
    a.deposit(-10)
  File "21801.py", line 10, in deposit
    assert amount > 0
AssertionError
```

□□□□□ assert □□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□

- □□□□□□
- □□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□

- [illegible]
- [illegible]

([illegible]Python [illegible])

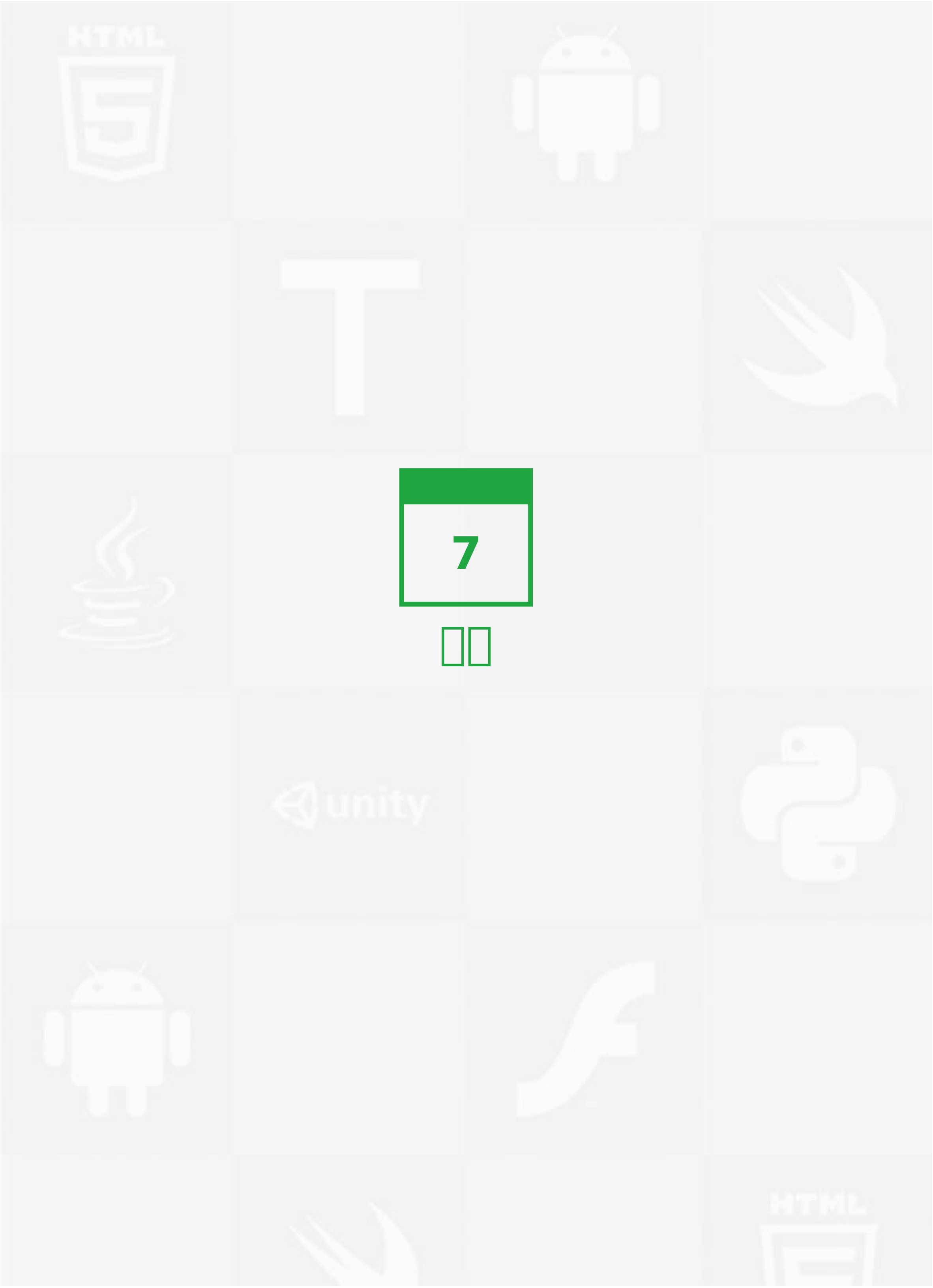
[illegible] master [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

> [illegible]
>
> [illegible]C [illegible] atoi [illegible]ASCII [illegible] 0[illegible] C [illegible] errno[illegible]
>
> [illegible]
>
> [illegible]
>
> [illegible] 0[illegible]
>
> [illegible]
>
> Python [illegible] try, except [illegible]

[illegible]

[illegible]qiwsir@126.com,**[illegible]**


HTML
5
unity

7

[illegible]

[illegible]

[illegible]Python [illegible] Python [illegible]“[illegible]”[illegible] Python [illegible]“[illegible]”[illegible] Python [illegible]

“[illegible]”[illegible]

Python [illegible]“[illegible]”[illegible]

> [illegible] 1865 [illegible]
>
> [illegible]
>
> [illegible] [illegible]

[illegible]

[illegible](1)[illegible] import [illegible]

```
>>> import math
>>> math.pow(3,2)
9.0
```

[illegible] math [illegible] import [illegible] pow() [illegible]

[illegible]

[illegible] `.py` [illegible] Python [illegible]

[illegible] `.py` [illegible] import [illegible] Python [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

lang = "python"
```

[illegible] pm.py[illegible] Python [illegible]

```
>>> import sys
>>>
sys.path.append("~/Documents/VBS/StartLearningPy
thon/2code/pm.py")
```

[illegible] Python [illegible] `import sys` [illegible] sys [illegible] Python [illegible] Python [illegible]

[illegible] ubuntu [illegible] windows [illegible]

```
>>> import pm
>>> pm.lang
'python'
```

[illegible] pm.py [illegible] `lang = "Python"` [illegible] `pm.py` [illegible]

```
>>> pm.xx
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'module' object has no attribute
'xx'
```

□□□□□□ pm.py □□□□□□□□□□□□□□□□□□□□□□□ .pyc □□□□□□□□□□□□□□□□□ □□□□□□□ Python□

> □□□□□□□□□□interpreter□□□□□□□□□□□□□□□ Python □□□□□□□□□□□ .py □□□□□□□□ .pyc □□□□□ .pyc □□□□□□□□□(bytecode)□□□□□□□□□□□□□□ .pyc □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ Python □□□ Python □□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□”□□

□□□□□□□□□“□□□□□□□□□□□□”□□□□□□□“□□□□”□□□□□□

□□□□□□□□□□□□□ .pyc □□□□□□□ Python □□□□□□□ .py □□□□□□□□□□□□□□□□□□□ .pyc □□□□□□□□□ .pyc □□□□□□□□□□ Python □□ .py □□□□□□□Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□[□□ Python □□□□□□□□]

□□□□□□ .pyc □□□□□□□□□□□□□□□□□□□□□□□□□□□□ .py □□□□□□□ .py □□□□□□□□□□□□Python □□□□□□□□□□□□□□□ .pyc □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `if __name__ == "__main__"` □□□□□□□□□ .py □□□□□□□□□□□□□□□□□□□□ .py □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ .py □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□ pm.py □□□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

def lang():
    return "Python"

if   name   == "__main__":
    print lang()
```

□□□□□□□□□□□□□□□□□

```
$ Python pm.py
python
```

□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> import sys
>>>
sys.path.append("~/Documents/VBS/StarterLearning
Python/2code/pm.py")
>>> import pm
>>> pm.lang()
'python'
```

□□□□□ pm.py □□□□ lang() □□□□□□□

```
>>> dir(pm)
['  builtins  ', '  doc  ', '  file__',
'__name__', '__package__', 'lang']
```

□□□□ .py □□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>>   name__
'__main__'
```

```
>>> pm.__name__
'pm'
```

□□□□□□□□□□□`__name__ == "__main__"`□□□□□□□□□□□`pm.__name__ == "pm"`□□□□`__name__`□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□`if __name__ == "__main__"`□

□□□□□

□□□□□□□□□□□□□□ Python □□□□□□□□□`sys.path.append("~/Documents/VBS/StarterLearningPython/2code/pm.py")`□□□□□ Python □□□□□□□□□□□□ sys.path □□□□□□□□□□□□□□□□□□□□□

```
>>> import sys
>>> import pprint
>>> pprint.pprint(sys.path)
['',
 '/usr/local/lib/python2.7/dist-
packages/autopep8-1.1-py2.7.egg',
 '/usr/local/lib/python2.7/dist-packages/pep8-
1.5.7-py2.7.egg',
 '/usr/lib/python2.7',
 '/usr/lib/python2.7/plat-i386-linux-gnu',
 '/usr/lib/python2.7/lib-tk',
 '/usr/lib/python2.7/lib-old',
 '/usr/lib/python2.7/lib-dynload',
 '/usr/local/lib/python2.7/dist-packages',
 '/usr/lib/python2.7/dist-packages',
 '/usr/lib/python2.7/dist-packages/PILcompat',
 '/usr/lib/python2.7/dist-packages/gtk-2.0',
 '/usr/lib/python2.7/dist-packages/ubuntu-sso-client',

'~/Documents/VBS/StarterLearningPython/2code/pm.py']
```

[illegible] .py [illegible] pm.py [illegible] pmlib.py[illegible] `'/usr/lib/Python2.7/dist-packages` [illegible] ubuntu [illegible]

```
 $ sudo cp pm.py /usr/lib/python2.7/dist-packages/pmlib.py
[sudo] password for qw:

$ ls /usr/lib/python2.7/dist-packages/pm*
/usr/lib/Python2.7/dist-packages/pmlib.py
```

[illegible]

```
 >>> import pmlib
>>> pmlib.lang
<function lang at 0xb744372c>
>>> pmlib.lang()
'python'
```

[illegible]——[illegible] sys.path [illegible]——[illegible] import [illegible]

PYTHONPATH [illegible]

[illegible] `sys.path.append()` [illegible] Python [illegible]

[illegible]

[illegible] PYTHONPATH [illegible]

> [illegible]

[illegible] ubuntu [illegible] Python [illegible] .py [illegible]

```
:~$ mkdir Python
:~$ cd python
:~/Python$ cp ~/Documents/VBS/StarterLearningPython/2code/pm.py mypm.py
:~/Python$ ls
mypm.py
```

[illegible] `~/Python` [illegible] `/home/qw/Python` [illegible]

```
vim /etc/profile
```

[illegible] root [illegible] `export PATH = /home/qw/python:$PAT` [illegible]

[illegible] `~/Python` [illegible] `Python` [illegible]

```
:~$ cd Python
:~/python$ Python

>>> import mypm
>>> mypm.lang()
'Python'
```

[illegible]

`__init__.py` [illegible]

`__init__.py` [illegible] .py [illegible] tornado [illegible](2)

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□□ (1)

“Python □□‘□□’”□□□□□□□□□□

□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□”□“□□”□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□“□□□□”□

□□□□□“□□□”□

□□□□□□□□□□□□□□□

□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

```
import modulename
```

□□□

```
>>> import pprint
>>> a = {"lang":"Python",
"book":"www.itdiffer.com", "teacher":"qiwsir",
"goal":"from beginner to master"}
>>> pprint.pprint(a)
{'book': 'www.itdiffer.com',
 'goal': 'from beginner to master',
 'lang': 'python',
 'teacher': 'qiwsir'}
```

[illegible] pprint [illegible] `pprint.pprint()` [illegible] dict [illegible]

[illegible] import [illegible]

[illegible] `import pprint` [illegible] `.` [illegible]

[illegible]

```
>>> from pprint import pprint
```

[illegible] `pprint` [illegible] `pprint()` [illegible]

```
>>> pprint(a)
{'book': 'www.itdiffer.com',
 'goal': 'from beginner to master',
 'lang': 'Python',
 'teacher': 'qiwsir'}
```

[illegible]

```
>>> from pprint import *
```

[illegible] pprint [illegible]

[illegible] `from modulename import name1, name2, name3...` [illegible]

[illegible]

```
>>> import pprint as pr
>>> pr.pprint(a)
{'book': 'www.itdiffer.com',
 'goal': 'from beginner to master',
 'lang': 'python',
 'teacher': 'qiwsir'}
```

[illegible]

```
>>> from pprint import pprint as pt
>>> pt(a)
{'book': 'www.itdiffer.com',
 'goal': 'from beginner to master',
 'lang': 'python',
 'teacher': 'qiwsir'}
```

[illegible]“[illegible]”[illegible]

[illegible]

[illegible] pprint [illegible]

```
>>> import pprint
>>> dir(pprint)
['PrettyPrinter', ' StringIO', '  all  ',
'  builtins  ', '  doc  ', '  file  ',
'  name  ', '  package  ', ' commajoin', '_id',
' len', ' perfcheck', ' recursion',
' safe repr', ' sorted', ' sys', ' type',
'isreadable', 'isrecursive', 'pformat',
'pprint', 'saferepr', 'warnings']
```

[illegible] dir() [illegible] pprint [illegible]

```
>>> [ m for m in dir(pprint) if not
m.startswith(' ') ]
['PrettyPrinter', 'isreadable', 'isrecursive',
'pformat', 'pprint', 'saferepr', 'warnings']
```

[illegible] `help()` [illegible]

```
>>> help(isreadable)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'isreadable' is not defined
```

[illegible]

```
>>> help(pprint.isreadable)
```

[illegible] `import pprint` [illegible]

```
Help on function isreadable in module pprint:

isreadable(object)
    Determine if saferepr(object) is readable by
eval().
```

[illegible]

[illegible] `pprint.PrettyPrinter` [illegible]

[illegible] `dir(pprint)` [illegible]

```
>>> pprint.  all
['pprint', 'pformat', 'isreadable',
'isrecursive', 'saferepr', 'PrettyPrinter']
```

[illegible]"warnings"[illegible]

[illegible] `from pprint import *` [illegible] `__all__` [illegible]

[illegible]

[illegible] dir() [illegible] help() [illegible]

```
>>> print pprint.  doc
Support to pretty-print lists, tuples, &
dictionaries recursively.

Very simple, but useful, especially in debugging
data structures.

Classes
-------

PrettyPrinter()
    Handle pretty-printing operations onto a
stream using a configured
    set of formatting parameters.

Functions
---------

pformat()
    Format a Python object into a pretty-printed
representation.

pprint()
```

```
    Pretty-print a Python object to a stream
[default is sys.stdout].

saferepr()
    Generate a 'standard' repr()-like value, but
protect against recursive
    data structures.
```

`pprint.__doc__`□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□(5)□□□□□□□□□□□□□□□□□□□□

□□□□ pm.py □□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

"""                                          #□□
□
This is a document of the python module.     #□□
□
"""                                          #□□
□

def lang():
    ...                                      #□□
□□□□□□□□□
```

□□□□□□□□□□□□□□□□□□□□□ import □□□□□□□□□□□□□□□□□□□□□□□
□□

```
>>> import sys
>>>
sys.path.append("~/Documents/VBS/StarterLearning
Python/2code")
>>> import pm
>>> print pm.__doc__
```

```
This is a document of the python module.
```

[illegible] .py [illegible]

Python [illegible]

[illegible] `dir(pprint)` [illegible] `__file__` [illegible]

```
>>> print pprint.  file
/usr/lib/python2.7/pprint.pyc
```

[illegible] ubuntu [illegible]

[illegible] .pyc [illegible] .py [illegible]

```
$ ls /usr/lib/python2.7/pp*
/usr/lib/python2.7/pprint.py
/usr/lib/python2.7/pprint.pyc
```

[illegible] pprint.py[illegible]

```
$ cat /usr/lib/python2.7/pprint.py

...

"""Support to pretty-print lists, tuples, &
dictionaries recursively.

Very simple, but useful, especially in debugging
data structures.

Classes
-------

PrettyPrinter()
    Handle pretty-printing operations onto a
```

```
stream using a configured
    set of formatting parameters.

Functions
---------

pformat()
    Format a Python object into a pretty-printed
representation.

....
"""
```

[illegible]`__doc__`[illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible] (2)

Python [illegible]

sys

[illegible] Python [illegible] `sys.path.append()` [illegible]

```
>>> import sys
>>> print sys.__doc__
```

[illegible] sys [illegible]

```
This module provides access to some objects used or maintained by the
interpreter and to functions that interact strongly with the interpreter.
```

[illegible] sys [illegible]

sys.argv

sys.argv [illegible] Python [illegible]“[illegible]”[illegible]

[illegible]

```
 $ Python --version
Python 2.7.6
```

□□□`--version` □□□□□□□□□□□□□ `Python --help` □□□□□□□

```
 $ Python --help
usage: Python [option] ... [-c cmd | -m mod |
file | -] [arg] ...
Options and arguments (and corresponding
environment variables):
-B     : don't write .py[co] files on import;
also PYTHONDONTWRITEBYTECODE=x
-c cmd : program passed in as string (terminates
option list)
-d     : debug output from parser; also
PYTHONDEBUG=x
-E     : ignore PYTHON* environment variables
(such as PYTHONPATH)
-h     : print this help message and exit (also
--help)
-i     : inspect interactively after running
script; forces a prompt even
         if stdin does not appear to be a
terminal; also PYTHONINSPECT=x
-m mod : run library module as a script
(terminates option list)
-O     : optimize generated bytecode slightly;
also PYTHONOPTIMIZE=x
-OO    : remove doc-strings in addition to the -
O optimizations
-R     : use a pseudo-random salt to make hash()
values of various types be
         unpredictable between separate
invocations of the interpreter, as
         a defense against denial-of-service
attacks
```

[illegible] `-B, -h` [illegible] `Python -h`[illegible] `-h` [illegible]

`sys.arg` [illegible] Python [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import sys

print "The file name: ", sys.argv[0]
print "The number of argument", len(sys.argv)
print "The argument is: ", str(sys.argv)
```

[illegible] 22101.py[illegible]

```
$ python 22101.py
The file name:  22101.py
The number of argument 1
The argument is:  ['22101.py']
```

[illegible]

- [illegible] `$ Python 22101.py` [illegible]“22101.py”[illegible]`Python` [illegible] `Python -h` [illegible] `-h` [illegible]
- sys.argv[0] [illegible] `22101.py`[illegible]

[illegible]

```
$ python 22101.py beginner master
www.itdiffer.com
The file name:  22101.py
The number of argument 4
The argument is:  ['22101.py', 'beginner',
'master', 'www.itdiffer.com']
```

[illegible] `sys.arg[1]` [illegible] `beginner` [illegible]

sys.exit()

[illegible]

```
Help on built-in function exit in module sys:

exit(...)
    exit([status])

    Exit the interpreter by raising
SystemExit(status).
    If the status is omitted or None, it
defaults to zero (i.e., success).
    If the status is an integer, it will be used
as the system exit status.
    If it is another kind of object, it will be
printed and the system
    exit status will be one (i.e., failure).
```

[illegible] `sys.exit()` [illegible] SystemExit [illegible] `os._exit()` [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import sys

for i in range(10):
    if i == 5:
        sys.exit()
    else:
        print i
```

[illegible]

```
 $ python 22102.py
0
1
2
3
4
```

[illegible] return[illegible] None[illegible] sys.exit() [illegible] SystemExit [illegible]

[illegible] `sys.exit(0)` [illegible] `sys.exit("I wet out at here.")`[illegible]

sys.path

`sys.path` [illegible] `append()` [illegible]

sys.stdin, sys.stdout, sys.stderr

[illegible] UNIX [illegible] Python [illegible]sys.stdin [illegible] raw_input() [illegible]Python3.x [illegible] imput()[illegible]sys.stdout [illegible]

> [illegible]([illegible])[illegible]stdin[illegible]stdout[illegible]

[illegible] `print()` [illegible] `sys.stdout.write(object + '\n')` [illegible]

```
>>> for i in range(3):
...     print i
...
0
1
2

>>> import sys
>>> for i in range(3):
...     sys.stdout.write(str(i))
...
012>>>
```

[illegible] `'\n'` [illegible]

```
>>> for i in range(3):
...     sys.stdout.write(str(i) + '\n')
...
0
1
2
```

[illegible] sys.stdout [illegible]“[illegible]”[illegible]“[illegible]”[illegible]

```
>>> f = open("stdout.md", "w")
>>> sys.stdout = f
>>> print "Learn Python: From Beginner to
Master"
>>> f.close()
```

[illegible] `sys.stdout = f` [illegible] print()[illegible]

[illegible]

```
$ cat stdout.md
Learn Python: From Beginner to Master
```

[illegible]

[illegible]

copy

[illegible](2)[illegible] copy [illegible]

```
>>> import copy
>>> copy.  all
['Error', 'copy', 'deepcopy']
```

[illegible] copy [illegible] deepcopy[illegible]

[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import copy

class MyCopy(object):
    def   init  (self, value):
        self.value = value

    def   repr  (self):
        return str(self.value)
```

```
foo = MyCopy(7)

a = ["foo", foo]
b = a[:]
c = list(a)
d = copy.copy(a)
e = copy.deepcopy(a)

a.append("abc")
foo.value = 17

print "original: %r\n slice: %r\n list(): %r\n copy(): %r\n deepcopy(): %r\n" % (a,b,c,d,e)
```

[illegible]

```
$ python 22103.py
original: ['foo', 17, 'abc']
 slice: ['foo', 17]
 list(): ['foo', 17]
 copy(): ['foo', 17]
 deepcopy(): ['foo', 7]
```

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□□ (3)

os

os □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> import os
>>> dir(os)
['EX CANTCREAT', 'EX CONFIG', 'EX DATAERR',
'EX IOERR', 'EX NOHOST', 'EX NOINPUT',
'EX NOPERM', 'EX NOUSER','EX OK', 'EX OSERR',
'EX OSFILE', 'EX PROTOCOL', 'EX SOFTWARE',
'EX TEMPFAIL', 'EX UNAVAILABLE', 'EX USAGE',
'F OK', 'NGROUPS MAX', 'O APPEND', 'O ASYNC',
'O CREAT', 'O DIRECT', 'O DIRECTORY', 'O_DSYNC',
'O EXCL', 'O LARGEFILE', 'O NDELAY',
'O NOATIME', 'O NOCTTY', 'O NOFOLLOW',
'O NONBLOCK', 'O RDONLY', 'O RDWR', 'O RSYNC',
'O SYNC', 'O TRUNC', 'O WRONLY', 'P NOWAIT',
'P NOWAITO', 'P WAIT', 'R OK', 'SEEK CUR',
'SEEK END', 'SEEK SET', 'ST APPEND',
'ST MANDLOCK', 'ST NOATIME', 'ST NODEV',
'ST NODIRATIME', 'ST NOEXEC', 'ST NOSUID',
'ST RDONLY', 'ST RELATIME', 'ST SYNCHRONOUS',
'ST WRITE', 'TMP MAX', 'UserDict', 'WCONTINUED',
'WCOREDUMP', 'WEXITSTATUS', 'WIFCONTINUED',
'WIFEXITED', 'WIFSIGNALED', 'WIFSTOPPED',
'WNOHANG', 'WSTOPSIG', 'WTERMSIG', 'WUNTRACED',
'W OK', 'X OK', ' Environ', '  all  ',
'__builtins__', '__doc__', '__file__',
'__name__', '__package__', '_copy_reg',
'_execvpe', '_exists', '_exit',
'_get_exports_list', '_make_stat_result',
'_make_statvfs_result', '_pickle_stat_result',
'_pickle_statvfs_result', '_spawnvef', 'abort',
'access', 'altsep', 'chdir', 'chmod', 'chown',
'chroot', 'close', 'closerange', 'confstr',
'confstr_names', 'ctermid', 'curdir', 'defpath',
'devnull', 'dup', 'dup2', 'environ', 'errno',
'error', 'execl', 'execle', 'execlp', 'execlpe',
'execv', 'execve', 'execvp', 'execvpe',
'extsep', 'fchdir', 'fchmod', 'fchown',
'fdatasync', 'fdopen', 'fork', 'forkpty',
'fpathconf', 'fstat', 'fstatvfs', 'fsync',
'ftruncate', 'getcwd', 'getcwdu', 'getegid',
'getenv', 'geteuid', 'getgid', 'getgroups',
'getloadavg', 'getlogin', 'getpgid', 'getpgrp',
'getpid', 'getppid', 'getresgid', 'getresuid',
'getsid', 'getuid', 'initgroups', 'isatty',
'kill', 'killpg', 'lchown', 'linesep', 'link',
'listdir', 'lseek', 'lstat', 'major', 'makedev',
'makedirs', 'minor', 'mkdir', 'mkfifo', 'mknod',
'name', 'nice', 'open', 'openpty', 'pardir',
'path', 'pathconf', 'pathconf_names', 'pathsep',
'pipe', 'popen', 'popen2', 'popen3', 'popen4',
'putenv', 'read', 'readlink', 'remove',
'removedirs', 'rename', 'renames', 'rmdir',
'sep', 'setegid', 'seteuid', 'setgid',
'setgroups', 'setpgid', 'setpgrp', 'setregid',
'setresgid', 'setresuid', 'setreuid', 'setsid',
'setuid', 'spawnl', 'spawnle', 'spawnlp',
'spawnlpe', 'spawnv', 'spawnve', 'spawnvp',
'spawnvpe', 'stat', 'stat_float_times',
'stat_result', 'statvfs', 'statvfs_result',
'strerror', 'symlink', 'sys', 'sysconf',
'sysconf_names', 'system', 'tcgetpgrp',
'tcsetpgrp', 'tempnam', 'times', 'tmpfile',
'tmpnam', 'ttyname', 'umask', 'uname', 'unlink',
'unsetenv', 'urandom', 'utime', 'wait', 'wait3',
'wait4', 'waitpid', 'walk', 'write']
```

[illegible] `os.popen()` [illegible] os [illegible] popen2[illegible]popen3[illegible]popen4[illegible] `help()` [illegible]——google[illegible] google[illegible]

[illegible]

[illegible]`open()` [illegible] os [illegible]

[illegible] 22201.py[illegible]

```
#!/usr/bin/env python
# coding=utf-8

print "This is a tmp file."
```

[illegible]

```
>>> import os
>>> os.rename("22201.py", "newtemp.py")
```

[illegible] 22201.py [illegible] Python [illegible]`os.rename("22201.py", "newtemp.py")`[illegible]

```
$ ls new*
newtemp.py
```

□□□□□□□□□□□□□□□□□□□□□ `cat newtemp.py` □□□□□□ ubuntu □□□□□□ windows □□□□□□□□□□□□□□□□□□□□□□

```
Help on built-in function rename in module posix:

rename(...)
    rename(old, new)

    Rename a file or directory.
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□ os.remove()□□□□□□□□□□□□□□□

```
Help on built-in function remove in module posix:

remove(...)
    remove(path)

    Remove a file (same as unlink(path)).
```

□□□□□□□□□□□□□□□□□□□□□□□□□□

```
$ pwd
/home/qw/Documents/VBS/StarterLearningPython/2code/rd
```

□□□□□□□□□□□□□□□□□□□

```
$ ls
a.py  b.py  c.py
```

□□□□ a.py □□

```
>>> import os
>>> 
os.remove("/home/qw/Documents/VBS/StarterLearningPython/2code/rd/a.py")
```

[illegible]

```
$ ls
b.py  c.py
```

[illegible]

```
>>> 
os.remove("/home/qw/Documents/VBS/StarterLearningPython/2code/rd")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
OSError: [Errno 21] Is a directory: 
'/home/qw/Documents/VBS/StarterLearningPython/2code/rd'
```

[illegible] `Remove a file`[illegible]os.remove() [illegible]

[illegible]

[illegible]

os.listdir[illegible]

```
Help on built-in function listdir in module posix:

listdir(...)
    listdir(path) -> list_of_strings
```

```
Return a list containing the names of the
entries in the directory.

    path: path of directory to list

The list is in arbitrary order.  It does not
include the special
entries '.' and '..' even if they are present in
the directory.
```

[illegible] linux [illegible] ls [illegible]

```
>>>
os.listdir("/home/qw/Documents/VBS/StarterLearni
ngPython/2code/rd")
['b.py', 'c.py']
>>> files =
os.listdir("/home/qw/Documents/VBS/StarterLearni
ngPython/2code/rd")
>>> for f in files:
...     print f
...
b.py
c.py
```

os.getcwd, os.chdir[illegible]

[illegible] `help()` [illegible]

```
>>> cwd = os.getcwd()     #[illegible]
>>> print cwd
/home/qw/Documents/VBS/StarterLearningPython/2co
de/rd
>>> os.chdir(os.pardir)    #[illegible]
>>> os.getcwd()             #□□
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode'

>>> os.chdir("rd")          #□□□□

>>> os.getcwd()
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode/rd'
```

`os.pardir` □□□□□□□□□□□□□□`..`

```
>>> os.pardir
'..'
```

os.makedirs, os.removedirs□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> dir = os.getcwd()
>>> dir
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode/rd'
>>> os.removedirs(dir)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib/python2.7/os.py", line 170, in
removedirs
    rmdir(name)
OSError: [Errno 39] Directory not empty:
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode/rd'
```

□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> os.getcwd()
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode'
```

[illegible]

```
>>> os.makedirs("newrd")
>>> os.chdir("newrd")
>>> os.getcwd()
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode/newrd'
```

[illegible]

```
>>> os.listdir(os.getcwd())
[]
>>> newdir = os.getcwd()
>>> os.removedirs(newdir)
```

[illegible] python [illegible]

[illegible]

```
>>> os.getcwd()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
OSError: [Errno 2] No such file or directory
```

[illegible]

```
>>> os.chdir(os.pardir)
>>> os.getcwd()
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode'
```

□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□ Python □□□□□

□□□□□□□□□□□□□□□□□□□□ `os.removedirs()` □□□□□□□□□□□
shutil □ retree □□□

```
 >>> os.getcwd()
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode'
>>> os.chdir("rd")
>>> now = os.getcwd()
>>> now
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode/rd'
>>> os.listdir(now)
['b.py', 'c.py']
>>> import shutil
>>> shutil.rmtree(now)
>>> os.getcwd()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
OSError: [Errno 2] No such file or directory
```

□□□□□□□□□□ os.makedirs() □□□□□□□□

```
 >>> os.getcwd()
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode'
>>> d0 = os.getcwd()
>>> d1 = d0+"/ndir1/ndir2/ndir3"    #□□□□□□□□□□□
□□□ ndir1,ndir2 □□□□□□
>>> d1
'/home/qw/Documents/VBS/StarterLearningPython/2c
ode/ndir1/ndir2/ndir3'
>>> os.makedirs(d1)
>>> os.chdir(d1)
>>> os.getcwd()
```

```
'/home/qw/Documents/VBS/StarterLearningPython/2code/ndir1/ndir2/ndir3'
```

[illegible] os.makedirs() [illegible] os.mkdir()[illegible] `os.mkdir()` [illegible]

`os.removedirs()` [illegible] `os.rmdir()` [illegible]

[illegible]

[illegible] os [illegible] `os.stat()`

```
>>> p = os.getcwd()     #[illegible]
>>> p
'/home/qw/Documents/VBS/StarterLearningPython'

# [illegible]
>>> os.stat(p)
posix.stat result(st mode=16895, st ino=4L,
st dev=26L, st nlink=1, st uid=0, st gid=0,
st size=12288L, st atime=1430224935,
st_mtime=1430224935, st_ctime=1430224935)

# [illegible]
>>> pf = p + "/README.md"
# [illegible]
>>> os.stat(pf)
posix.stat result(st mode=33279, st ino=67L,
st dev=26L, st nlink=1, st uid=0, st_gid=0,
st size=50L, st atime=1429580969,
st_mtime=1429580969, st_ctime=1429580969)
```

[illegible] computer [illegible]

```
>>> fi = os.stat(pf)
>>> mt = fi[8]
```

fi[8] □□ st_mtime □□□□□□□□ modified□□□□□□□□□□□□□□

```
>>> mt
1429580969
```

□□□□□□□□□□ time □□□□□□□□

```
>>> import time
>>> time.ctime(mt)
'Tue Apr 21 09:49:29 2015'
```

□□□□□□□□□□

□ `os.stat()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□ Python □ os □□□□□□□

□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□ □□ os □□□□□ `os.chmod()`

□□□□

□□□□□□□□ linux □□□□□□□□□ dos□□□□□□□□□□ windows □□□□ command□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ ubuntu □□□ □□□□□□□□□□□ `ls` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□os □□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□ ubuntu □□□□□□□□ windows□□□□□□□□ DOS □□□□

```
>>> p
'/home/qw/Documents/VBS/StarterLearningPython'
>>> command = "ls " + p
>>> command
```

```
'ls
/home/qw/Documents/VBS/StarterLearningPython'
```

[illegible] command[illegible]

```
>>> os.system(command)
01.md     101.md  105.md  109.md  113.md  117.md
121.md  125.md  129.md   201.md  205.md  209.md
213.md  217.md  221.md   index.md
02.md     102.md  106.md  110.md  114.md  118.md
122.md  126.md  130.md   202.md  206.md  210.md
214.md  218.md  222.md   n001.md
03.md     103.md  107.md  111.md  115.md  119.md
123.md  127.md  1code      203.md  207.md
211.md  215.md  219.md  2code    README.md
0images  104.md  108.md  112.md  116.md  120.md
124.md  128.md  images  204.md  208.md  212.md
216.md  220.md  images
0
```

[illegible]

[illegible] `os.system()` [illegible] `os.exec` [illegible] `os.execvp` [illegible] `os.system()` [illegible] shell [illegible] `os.exec()` [illegible] Python [illegible]

[illegible] Python [illegible]

`os.system()` [illegible]

```
>>> os.system("/usr/bin/firefox")

(process:4002): GLib-CRITICAL **:
```

```
g_slice_set_config: assertion 'sys_page_size == 0' failed

(firefox:4002): GLib-GObject-WARNING **: Attempt to add property GnomeProgram::sm-connect after class was initialised
......
```

[illegible] ubuntu [illegible] `/usr/bin/firefox` [illegible] windows[illegible] windows [illegible] Python [illegible] `\` [illegible](1)[illegible](2)[illegible] `r"c:\user\firfox.exe"` [illegible] `r"  "` [illegible] windows [illegible] `C:\Program Files` [illegible] `os.system()` [illegible] firefox [illegible]

[illegible] webbrowser [illegible]

```
>>> import webbrowser
>>> webbrowser.open("http://www.itdiffer.com")
True
```

[illegible]

[illegible]“[illegible] Python”[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□□□ (4)

heapq

□□heap□□□□□□□□□□□□□□□□□□□□

> □□□□□heap)□□

□□

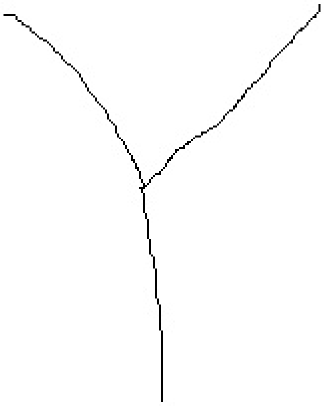
□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

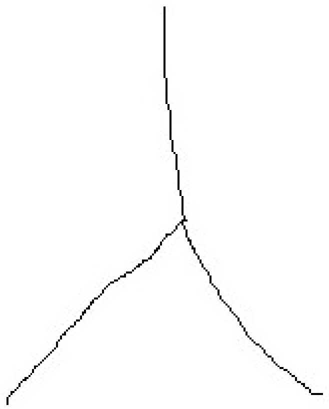
[illegible]

[illegible]“[illegible]”[illegible]

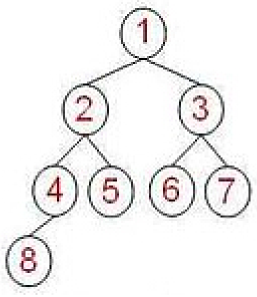


[illegible]“[illegible]”[illegible]

[illegible]



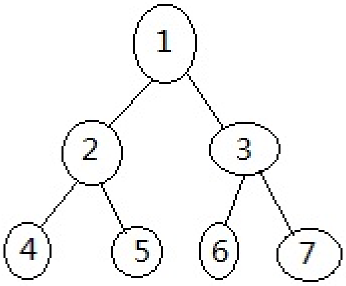
[illegible]



[illegible]“[illegible]”[illegible]

> [illegible]Binary tree[illegible]left subtree[illegible]right subtree[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]Complete Binary Tree[illegible]Full Binarry Tree)[illegible]



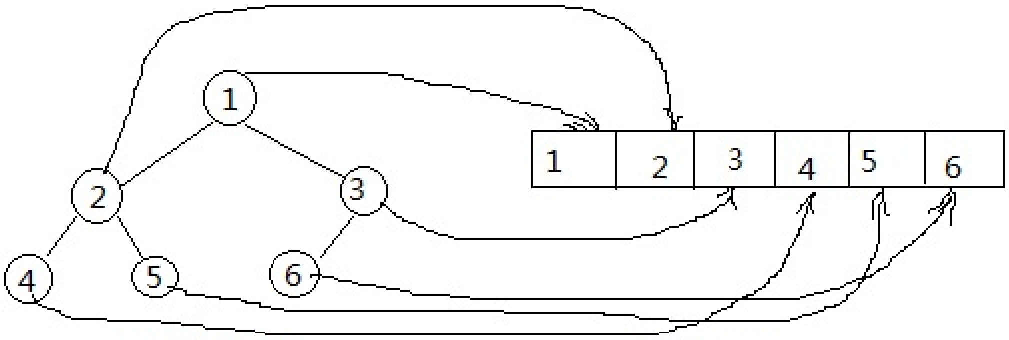
[illegible]

- [illegible]
- [illegible]

[illegible]

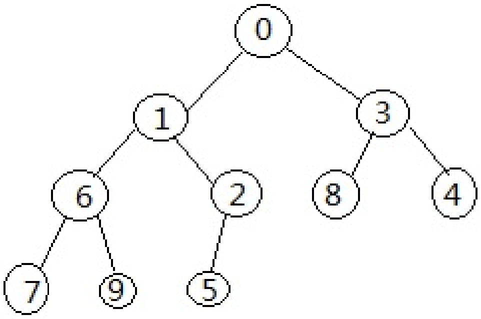
[illegible]

[illegible]


1
2
3
4
5
6
1
2
3
4
5
6

(a)逻辑结构 (b)存储结构

[illegible]



□□

□□
□□□□ Python □□□□ heapq □□□□□□□□

heapq □□

heapq □□ heap □□□q □□ queue□□□□□□□□□□□□□□

```
>>> import heapq
>>> heapq.__all__
['heappush', 'heappop', 'heapify',
'heapreplace', 'merge', 'nlargest', 'nsmallest',
'heappushpop']
```

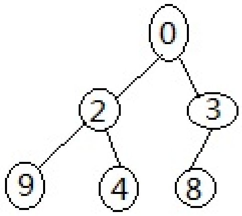
□□□□□□□□□□□□□□

heappush(heap, x)□□ x □□□ heap□□□□□□□□

```
Help on built-in function heappush in module
_heapq:

heappush(...)
    heappush(heap, item) -> None. Push item onto
heap, maintaining the heap invariant.
```

```
>>> import heapq
>>> heap = []
>>> heapq.heappush(heap, 3)
>>> heapq.heappush(heap, 9)
>>> heapq.heappush(heap, 2)
>>> heapq.heappush(heap, 4)
>>> heapq.heappush(heap, 0)
>>> heapq.heappush(heap, 8)
>>> heap
[0, 2, 3, 9, 4, 8]
```

[illegible]



[illegible] `heappush()` [illegible]

heappop(heap)[illegible]

[illegible]

```
>>> heapq.heappop(heap)
0
>>> heap
[2, 4, 3, 9, 8]
```

[illegible] `heappop()` [illegible] heap [illegible] heap [illegible] 0 [illegible]

heapify()[illegible]

□□□□□□□□□□□□□□ `heapify()` □□□□□□□□□□□□

```
>>> hl = [2, 4, 6, 8, 9, 0, 1, 5, 3]
>>> heapq.heapify(hl)
>>> hl
[0, 3, 1, 4, 9, 6, 2, 5, 8]
```

□□□□□□□□□□ hl □□□□□□□□□□□□□□□□□□□□□□□□□ hl□□□□□
heappop() □□ heappush() □□□□□□□□□□□

```
>>> heapq.heappop(hl)
0
>>> heapq.heappop(hl)
1
>>> hl
[2, 3, 5, 4, 9, 6, 8]
>>> heapq.heappush(hl, 9)
>>> hl
[2, 3, 5, 4, 9, 6, 8, 9]
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> heapq.heappush(hl, "q")
>>> hl
[2, 3, 5, 4, 9, 6, 8, 9, 'q']
>>> heapq.heappush(hl, "w")
>>> hl
[2, 3, 5, 4, 9, 6, 8, 9, 'q', 'w']
```

heapreplace()

□ heappop() □ heappush() □□□□□□□□□□□□□□□□□□□□□□

```
>>> heap
[2, 4, 3, 9, 8]
>>> heapq.heapreplace(heap, 3.14)
2
```

```
>>> heap
[3, 4, 3.14, 9, 8]
```

[illegible] sorted() [illegible] sort()[illegible]

[illegible]

deque [illegible]

[illegible]`[1,2,3]`[illegible]

[illegible] `append()` [illegible]

[illegible]

[illegible]

```
>>> lst = [1, 2, 3]
>>> lst.append(4)
>>> lst
[1, 2, 3, 4]
>>> nl = [7]
>>> nl.extend(lst)
>>> nl
[7, 1, 2, 3, 4]
```

[illegible]Python [illegible]

```
>>> from collections import deque
```

[illegible] collections [illegible] deque[illegible]

```
>>> lst
[1, 2, 3, 4]
```

□□□□□□□□□□□□□□□□□□□□

```
>>> qlst = deque(lst)
```

□□□□□□□□□□□□ deque□deque □□□□□□□□□□□□“□□□□”□double-ended queue□□

```
>>> qlst.append(5)        #□□□□□
>>> qlst
deque([1, 2, 3, 4, 5])
>>> qlst.appendleft(7)    #□□□□□
>>> qlst
deque([7, 1, 2, 3, 4, 5])
```

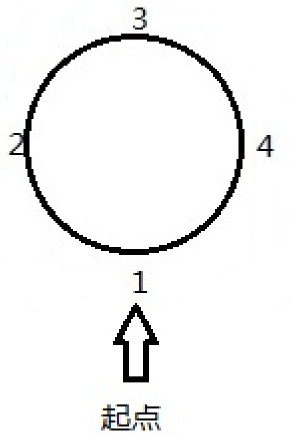
□□□□□□□□□□□□□□□□

```
>>> qlst.pop()
5
>>> qlst
deque([7, 1, 2, 3, 4])
>>> qlst.popleft()
7
>>> qlst
deque([1, 2, 3, 4])
```

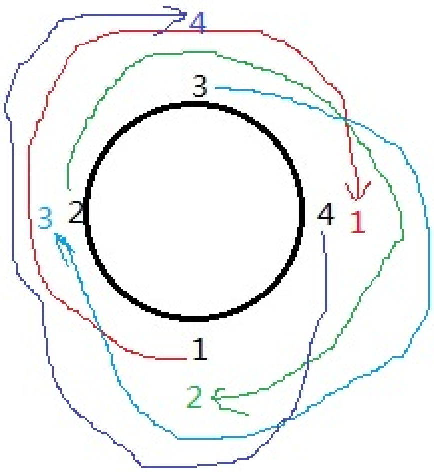
□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> qlst.rotate(3)
>>> qlst
deque([2, 3, 4, 1])
```

rotate() □□□□□[1, 2, 3, 4]□□□□□□□□□□□□□□□□□□□ 1,2,3,4 □□□□□□□□□□□□□□ 1□□□□□□□□□□□□□ 1 □□□□□□□□□□□□



□□ `rotate()` □□□□□□□□□□□□□□ `rotate(3)` □□□□□□□□□□□□□□
□□□□□□□□□□□□□



□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□[2, 3, 4, 1]□rotate()
□□□□□□□□□□□□□

```
 >>> qlst
deque([3, 4, 1, 2])
>>> qlst.rotate(-1)
>>> qlst
deque([4, 1, 2, 3])
```

□□□□□□□□□□□□□□□□

□ deque □□□□ extend □ extendleft □□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

□□□ (5)

“□□□□□□□□□□□□□□□”□□□□□□□□

□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□□□□□
Python □□“□□”□□□□□□□□□□□

calendar

```
>>> import calendar
>>> cal = calendar.month(2015, 1)
>>> print cal
    January 2015
Mo Tu We Th Fr Sa Su
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

□□□□□□□ 2015 □ 1 □□□□□□□□□□□□□□□□□□□ calendar □□□□□□□□ dir() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

calendar(year,w=2,l=1,c=6)

□□ year □□□□3 □□□□□□□□□□ c□ □□□□□□□ w □□□□□□□□ 21* W+18+2* C□l □□□□□□□

```
>>> year = calendar.calendar(2015)
>>> print year
```

```
                                  2015

      January                       February
March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
Mo Tu We Th Fr Sa Su
          1  2  3  4                         1
1
 5  6  7  8  9 10 11       2  3  4  5  6  7  8
2  3  4  5  6  7  8
12 13 14 15 16 17 18       9 10 11 12 13 14 15
9 10 11 12 13 14 15
19 20 21 22 23 24 25      16 17 18 19 20 21 22
16 17 18 19 20 21 22
26 27 28 29 30 31         23 24 25 26 27 28
23 24 25 26 27 28 29

30 31

       April                          May
June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
Mo Tu We Th Fr Sa Su
       1  2  3  4  5                   1  2  3
1  2  3  4  5  6  7
 6  7  8  9 10 11 12       4  5  6  7  8  9 10
8  9 10 11 12 13 14
13 14 15 16 17 18 19      11 12 13 14 15 16 17
15 16 17 18 19 20 21
20 21 22 23 24 25 26      18 19 20 21 22 23 24
22 23 24 25 26 27 28
27 28 29 30               25 26 27 28 29 30 31
29 30

        July                         August
September
```

```
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
Mo Tu We Th Fr Sa Su
       1  2  3  4  5                      1  2
1  2  3  4  5  6
 6  7  8  9 10 11 12       3  4  5  6  7  8  9
7  8  9 10 11 12 13
13 14 15 16 17 18 19      10 11 12 13 14 15 16
14 15 16 17 18 19 20
20 21 22 23 24 25 26      17 18 19 20 21 22 23
21 22 23 24 25 26 27
27 28 29 30 31            24 25 26 27 28 29 30
28 29 30
                          31

      October                   November
December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
Mo Tu We Th Fr Sa Su
          1  2  3  4                         1
1  2  3  4  5  6
 5  6  7  8  9 10 11       2  3  4  5  6  7  8
7  8  9 10 11 12 13
12 13 14 15 16 17 18       9 10 11 12 13 14 15
14 15 16 17 18 19 20
19 20 21 22 23 24 25      16 17 18 19 20 21 22
21 22 23 24 25 26 27
26 27 28 29 30 31         23 24 25 26 27 28 29
28 29 30 31
                          30
```

isleap(year)

□□□□□□□□□□□□ true□□□ false.

```
 >>> calendar.isleap(2000)
True
```

```
>>> calendar.isleap(2015)
False
```

[illegible]

leapdays(y1,y2)

[illegible] Y1[illegible]Y2 [illegible] y1[illegible] y2[illegible]

```
 >>> calendar.leapdays(2000,2004)
1
>>> calendar.leapdays(2000,2003)
1
```

month(year,month,w=2,l=1)

[illegible] year [illegible] month [illegible] w [illegible] 7*
w+6[illegible]l [illegible]

```
 >>> print calendar.month(2015, 5)
      May 2015
Mo Tu We Th Fr Sa Su
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

monthcalendar(year,month)

[illegible]
[illegible] 0[illegible]

```
 >>> calendar.monthcalendar(2015, 5)
[[0, 0, 0, 0, 1, 2, 3], [4, 5, 6, 7, 8, 9, 10],
[11, 12, 13, 14, 15, 16, 17], [18, 19, 20, 21,
22, 23, 24], [25, 26, 27, 28, 29, 30, 31]]
```

□□□□□□□□□□ `calendar.month(2015, 5)` □□□□□□

monthrange(year,month)

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 0 □□□□□□□□□□□ □□□□□ 6 □□□□□□□□□□□□□□□□□□□□□

```
>>> calendar.monthrange(2015, 5)
(4, 31)
```

□□□□□□□2015 □ 5 □ 1 □□□□□□□□□□□ 31 □□□□□□□□□□□□□□□□□

weekday(year,month,day)

□□□□□□□□□□□□□□□□□□□□□□□□□□ 0 □ 6 □□□□□□□□□□□□□

```
>>> calendar.weekday(2015, 5, 4)    #□□□
0
>>> calendar.weekday(2015, 6, 4)    #□□□
3
```

time

time()

time □□□□□□□

```
>>> import time
>>> time.time()
1430745298.391026
```

`time.time()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1970 □ 1 □ 1 □ 0 □ 0 □ 0 □□□□□□□□□□□□□□□□□□□□□

> UNIX □□□□□ POSIX □□□ UNIX □□ UNIX □□□□□□□□□□□□□□□□□
> □ 1970 □ 1 □ 1 □ 0 □ 0 □ 0 □□□□□□□□□□□□□□
>
> □□□□□□□ UNIX □□□□□ 32 □□□□□□□□ 32 □□□□□□□□□□□□□□□
> □□□□□□□□□□□□□ 2038 □ 1 □ 19 □ 3 □ 14 □ 07 □□□□□□
> 01111111 11111111 11111111 11111111□
> 0x7FFF:FFFF□□□□□□□□□□□□□ 10000000 00000000
> 00000000 00000000□□0x8000:0000□□□□□□□□□□□□□□□□□
> □□ 1901 □ 12 □ 13 □ 20 □ 45 □ 52 □□□□□□□□ 1970 □□□□□□
> □□□□□□□□□□□□□□□□□
>
> □□□□□□□□□□□□ 32 □□□□ 64 □□□□□□ 64 □□□□□□□□□□□□□□□
> 292,277,026,596 □ 12 □ 4 □ 15 □ 30 □ 08 □□

□□□□□□□□□□□□□□□□

localtime()

```
>>> time.localtime()
time.struct_time(tm_year=2015, tm_mon=5,
tm_mday=4, tm_hour=21, tm_min=33, tm_sec=39,
tm_wday=0, tm_yday=124, tm_isdst=0)
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□	□□	□□
0	tm_year	□
1	tm_mon	□
2	tm_mday	□
3	tm_hour	□
4	tm_min	□

[illegible]	[illegible]	[illegible]
5	tm_sec	[illegible]
6	tm_wday	[illegible]
7	tm_yday	[illegible]
8	tm_isdst	[illegible]

```
>>> t = time.localtime()
>>> t[1]
5
```

[illegible]

[illegible] `time.localtime()` [illegible] `time.time()` [illegible]

```
>>> time.localtime(100000)
time.struct_time(tm_year=1970, tm_mon=1,
tm_mday=2, tm_hour=11, tm_min=46, tm_sec=40,
tm_wday=4, tm_yday=2, tm_isdst=0)
```

gmtime()

localtime() [illegible]

```
>>> import time
>>> time.gmtime()
time.struct_time(tm_year=2015, tm_mon=5,
tm_mday=4, tm_hour=23, tm_min=46, tm_sec=34,
tm_wday=0, tm_yday=124, tm_isdst=0)
```

> [illegible] Greenwich Mean Time [illegible] GMT [illegible]

> [illegible]

[illegible]

asctime()

```
>>> time.asctime()
'Mon May  4 21:46:13 2015'
```

`time.asctime()` [illegible] `time.localtime()` [illegible]

```
>>> h = time.localtime(1000000)
>>> h
time.struct time(tm year=1970, tm mon=1,
tm mday=12, tm hour=21, tm min=46, tm_sec=40,
tm wday=0, tm yday=12, tm_isdst=0)
>>> time.asctime(h)
'Mon Jan 12 21:46:40 1970'
```

[illegible] `time.asctime()` [illegible] `time.time()` [illegible]

ctime()

```
>>> time.ctime()
'Mon May  4 21:52:22 2015'
```

[illegible] `time.time()` [illegible]

```
>>> time.ctime(1000000)
'Mon Jan 12 21:46:40 1970'
```

[illegible]

[illegible] localtime()[illegible]gmtime() [illegible] time() [illegible] asctime() [illegible] ctime() [illegible]

mktime()

mktime() [illegible]

```
>>> lt = time.localtime()
>>> lt
time.struct time(tm year=2015, tm mon=5,
tm mday=5, tm hour=7, tm min=55, tm_sec=29,
tm wday=1, tm yday=125, tm_isdst=0)
>>> time.mktime(lt)
1430783729.0
```

[illegible] localtime() [illegible]localtime() [illegible]

[illegible]“[illegible]”[illegible]

strftime()

[illegible]

> Help on built-in function strftime in module time:
>
> strftime(...) strftime(format[, tuple]) -> string
> Convert a time tuple to a string according to a format specification. See the library reference manual for formatting codes. When the time tuple is not present, current time as returned by localtime() is used.

[illegible] localtime() [illegible] format[illegible]

[illegible]	[illegible]	[illegible]
%y	[illegible]	00-99[illegible]"15"
%Y	[illegible]	[illegible]"2015"

□□	□□	□□□□□□□□
%j	□□□□□□□□□□□□	001-366
%m	□□□□	01-12
%b	□□□□□□□□□	□□□□□□
%B	□□□□□□□□□	□□□□□□
%d	□□□□□□	□ 5 □ 1 □□□"01"
%H	□□□□□□□24 □□□□	00-23
%I	□□□□□□□12 □□□□	01-12
%M	□□	00-59
%S	□	00-59
%U	□□□□□□□□□□□□□□□□□□□□	00-53
%W	□□□□□□□□□□□□□	00-53
%w	□□□□□□□□	0-6
%Z	□□	□□□□□□□□□□ CST□□ China Standard Time
%x	□□	□/□/□
%X	□□	□:□:□
%c	□□□□□□	□/□/□ □:□:□
%%	‘%’□□	‘%’□□

□□	□□	□□□□□□□□
%p	□□□	AM or PM

□□□□□□□

```
>>> time.strftime("%y,%m,%d")
'15,05,05'
>>> time.strftime("%y/%m/%d")
'15/05/05'
```

□□□□□□□□□□□□□□□□□□□□□□□□“□□□□□□□”□□

strptime()

> Help on built-in function strptime in module time:
>
> strptime(...) strptime(string, format) -> struct_time
> Parse a string to a time tuple according to a format
> specification. See the library reference manual for
> formatting codes (same as strftime()).

strptime() □□

```
>>> today = time.strftime("%y/%m/%d")
>>> today
'15/05/05'
>>> time.strptime(today, "%y/%m/%d")
time.struct_time(tm_year=2015, tm_mon=5,
tm_mday=5, tm_hour=0, tm_min=0, tm_sec=0,
tm_wday=1, tm_yday=125, tm_isdst=-1)
```

datetime

□□ time □□ □□□□□□□ datetime □□□□□□□□□□□□□□□□□□

datetime □□□□□□□□

- datetime.date□□□□□□□□□□□ year/month/day
- datetime.time□□□□□□□□□ hour/minute/second/microsecond
- datetime.datetime□□□□□□
- datetime.timedelta□□□□□□□□□□□□□□□□□□□□
- datetime.tzinfo□□□□

date □

□□□□□□□□□□□□

```
>>> import datetime
>>> today = datetime.date.today()
>>> today
datetime.date(2015, 5, 5)
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ print □□□□□□□□□□□

```
>>> print today
2015-05-05
>>> print today.ctime()
Tue May  5 00:00:00 2015
>>> print today.timetuple()
time.struct_time(tm_year=2015, tm_mon=5,
tm_mday=5, tm_hour=0, tm_min=0, tm_sec=0,
tm_wday=1, tm_yday=125, tm_isdst=-1)
>>> print today.toordinal()
735723
```

[illegible] `datetime.date.year()` [illegible] year [illegible]
[illegible]

```
>>> print today.year
2015
>>> print today.month
5
>>> print today.day
5
```

[illegible]

```
>>> to = today.toordinal()
>>> to
735723
>>> print datetime.date.fromordinal(to)
2015-05-05

>>> import time
>>> t = time.time()
>>> t
1430787994.80093
>>> print datetime.date.fromtimestamp(t)
2015-05-05
```

[illegible]

```
>>> d1 = datetime.date(2015,5,1)
>>> print d1
2015-05-01
>>> d2 = d1.replace(year=2005, day=5)
>>> print d2
2005-05-05
```

time [illegible]

□□□□ time □□

```
>>> t = datetime.time(1,2,3)
>>> print t
01:02:03
```

□□□□□□□

```
>>> print t.hour
1
>>> print t.minute
2
>>> print t.second
3
>>> t.microsecond
0
>>> print t.tzinfo
None
```

timedelta □

□□□□□□□□□□□□□□□

```
>>> now = datetime.datetime.now()
>>> print now
2015-05-05 09:22:43.142520
```

□□□□ datetime □□□□□□□ date □ time □□□□□□□□□□□□□□□□□□□□□ now □□□□

□ now □□ 5 □□□

```
>>> b = now + datetime.timedelta(hours=5)
>>> print b
2015-05-05 14:22:43.142520
```

□□□□

```
>>> c = now + datetime.timedelta(weeks=2)
>>> print c
2015-05-19 09:22:43.142520
```

□□□□□□

```
>>> d = c - b
>>> print d
13 days, 19:00:00
```

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

□□□ (6)

urllib

urllib □□□□□□□□□□□□□□□□□□□□□□□□□□ Python □□□□□□□□□□□ □□□□□□□□□□□□□□

```
>>> import urllib
>>> itdiffer =
urllib.urlopen("http://www.itdiffer.com")
```

□□□□□□□□□□ [www.itdiffer.comhref="http://www.itdiffer.com)□□□□□□□□ □□ □□□(1)□

```
>>> print itdiffer.read()
<!DOCTYPE HTML>
<html>
    <head>
        <title>I am Qiwsir</title>
....//□□□□□□□□□□□□□
```

□□□ □□□□□□□□□ urllib □□□□□□□□□□□

```
>>> dir(urllib)
['ContentTooShortError', 'FancyURLopener',
'MAXFTPCACHE', 'URLopener', '  all  ',
'  builtins  ', '  doc  ', '  file  ',
'  name  ', '  package  ', '  version  ',
' asciire', ' ftperrors', ' have ssl',
' hexdig', ' hextochr', ' hostprog',
' is unicode', ' localhost', ' noheaders',
' nportprog', ' passwdprog', ' portprog',
' queryprog', ' safe map', ' safe quoters',
' tagprog', ' thishost', ' typeprog',
' urlopener', ' userprog', ' valueprog',
'addbase', 'addclosehook', 'addinfo',
'addinfourl', 'always safe', 'base64',
'basejoin', 'c', 'ftpcache', 'ftperrors',
'ftpwrapper', 'getproxies',
'getproxies environment', 'i', 'localhost',
'noheaders', 'os', 'pathname2url',
'proxy bypass', 'proxy bypass environment',
'quote', 'quote plus', 're', 'reporthook',
'socket', 'splitattr', 'splithost',
'splitnport', 'splitpasswd', 'splitport',
'splitquery', 'splittag', 'splittype',
'splituser', 'splitvalue', 'ssl', 'string',
'sys', 'test1', 'thishost', 'time', 'toBytes',
'unquote', 'unquote plus', 'unwrap',
'url2pathname', 'urlcleanup', 'urlencode',
'urlopen', 'urlretrieve']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

urlopen()

urlopen() □□□□□□ url □□□□□□□□□□ url □□□□□□□□□□□□□□□□□□
□□□□□

> Help on function urlopen in module urllib:
>
> urlopen(url, data=None, proxies=None) Create a file-like object for the specified URL to read from.

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- url□□□□□□□□□□□□□□
- data□□□□□ post □□□□□□□□□□□□□
- proxies□□□□□

□□□□□□□□□□□□□□□Python□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□ itdiffer □□□□□□□□□□□□□□□□□□

```
>>> dir(itdiffer)
['  doc  ', '  init  ', '  iter  ',
'  module  ', '  repr  ', 'close', 'code',
'fileno', 'fp', 'getcode', 'geturl', 'headers',
'info', 'next', 'read', 'readline', 'readlines',
'url']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- read(),readline(),readlines(),fileno(),close()□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□
- info()□□□□□□
- getcode()□□□ http □□□
- geturl()□□□ url

□□□□□

```
>>> itdiffer.info()
<httplib.HTTPMessage instance at 0xb6eb3f6c>
>>> itdiffer.getcode()
200
>>> itdiffer.geturl()
'http://www.itdiffer.com'
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□ url □□□□□

url □□□□□□□□□□□□□□□□□□□□□□□□□□ url □□□□□□□□□□□□□□
web □□□□□□□□□□□urllib □□□□□□□□□

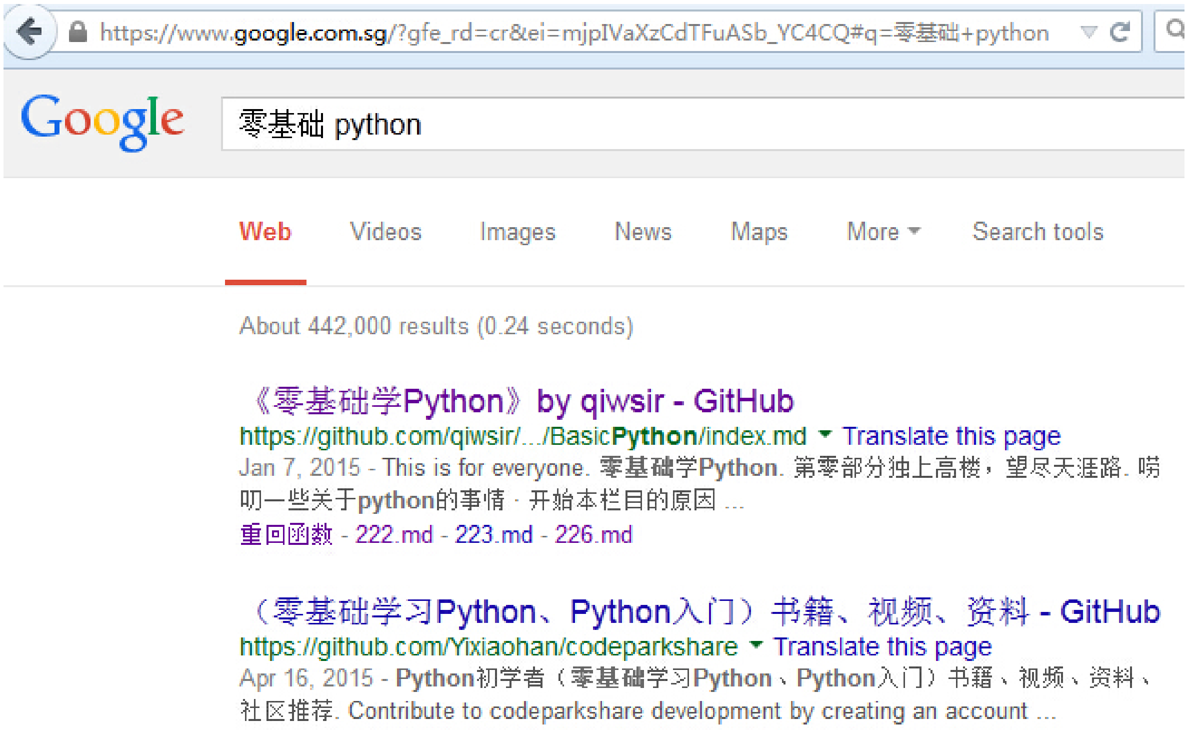
- quote(string[, safe])□□□□□□□□□□□□ safe □□□□□□□□□□□
- urllib.unquote(string) □□□□□□□□□
- quote_plus(string [, safe]) □□ urllib.quote □□□□□□□□ □'+'□□□□□`' '`□□ quote □'%20'□□□□□
- unquote_plus(string) □□□□□□□□□□
- urllib.urlencode(query[, doseq])□□ dict □□□□□□□□□□□□ □□□□ url □□□□□{'name': 'laoqi', 'age': 40}□□□□ □"name=laoqi&age=40"
- pathname2url(path)□□□□□□□□□ url □□
- url2pathname(path)□□ url □□□□□□□□□

□□□□□□□□□

```
>>> du = "http://www.itdiffer.com/name=python
book"
>>> urllib.quote(du)
'http%3A//www.itdiffer.com/name%3Dpython%20book'
>>> urllib.quote_plus(du)
'http%3A%2F%2Fwww.itdiffer.com%2Fname%3Dpython+b
ook'
```

□□□□□□□□□□□□□□□ `%20` □□□□□□ `+`

□□□□□□□□□ google □□□ `□□□ Python` □□□□□□□



https://www.google.com.sg/?gfe_rd=cr&ei=mjpIVaXzCdTFuASb_YC4CQ#q=零基础+python

零基础 python

About 442,000 results (0.24 seconds)

《零基础学Python》by qiwsir - GitHub
https://github.com/qiwsir/.../BasicPython/index.md Translate this page
Jan 7, 2015 - This is for everyone. 零基础学Python. 第零部分独上高楼，望尽天涯路. 唠叨一些关于python的事情 · 开始本栏目的原因 ...
重回函数 - 222.md - 223.md - 226.md

（零基础学习Python、Python入门）书籍、视频、资料 - GitHub
https://github.com/Yixiaohan/codeparkshare Translate this page
Apr 16, 2015 - Python初学者（零基础学习Python、Python入门）书籍、视频、资料、社区推荐. Contribute to codeparkshare development by creating an account ...

□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□ url□□□□□ `+` □□□□□□

```
>>> dup = urllib.quote_plus(du)
>>> urllib.unquote_plus(dup)
'http://www.itdiffer.com/name=Python book'
```

□□□□□□□□□□□□□□□□□

```
>>>
urllib.urlencode({"name":"qiwsir","web":"itdiffe
r.com"})
'web=itdiffer.com&name=qiwsir'
```

□□□□□□□□□□□□□□□□□□□□□□

urlretrieve()

□□ urlopen() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ urlretrieve() □□□□□□□□□□□□□□□□

```
>>> import urllib
>>>
urllib.urlretrieve("http://www.itdiffer.com/imag
es/me.jpg","me.jpg")
('me.jpg', <httplib.HTTPMessage instance at
0xb6ecb6cc>)
>>>
```

me.jpg [illegible]
http://www.itdiffer.com/images/me.jpg[illegible] me.jpg[illegible] Python [illegible]

[illegible]urllib[illegible]

```
urllib.urlretrieve(url[, filename[, reporthook[,
data]]])
```

- url[illegible]
- filename[illegible]urllib [illegible]
- reporthook[illegible]
- data[illegible] post [illegible]

[illegible](filename, headers)[illegible]filename [illegible]headers [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import urllib

def go(a,b,c):
    per = 100.0 * a * b / c
    if per > 100:
        per = 100
    print "%.2f%%" % per

url = "http://youxi.66wz.com/uploads/1046/1321/11410192.90d133701b06f0cc2826c3e5ac34c620.jpg"
local = "/home/qw/Pictures/g.jpg"
urllib.urlretrieve(url, local, go)
```

[illegible]

```
$ Python 22501.py
0.00%
8.13%
16.26%
24.40%
32.53%
40.66%
48.79%
56.93%
65.06%
73.19%
81.32%
89.46%
97.59%
100.00%
```

[illegible]

urllib2

urllib2 [illegible] urllib [illegible]——[illegible] url [illegible] [illegible]Python: difference between urllib and urllib2

□□□□□□□□□□□□□

> urllib2 can accept a Request object to set the headers for a URL request, urllib accepts only a URL. That means, you cannot masquerade your User Agent string etc.
>
> urllib provides the urlencode method which is used for the generation of GET query strings, urllib2 doesn't have such a function. This is one of the reasons why urllib is often used along with urllib2.

□□□□□□□□□□□□□□□urllib □□□ urllib2 □□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□

```
>>> dir(urllib2)
['AbstractBasicAuthHandler',
'AbstractDigestAuthHandler',
'AbstractHTTPHandler', 'BaseHandler',
'CacheFTPHandler', 'FTPHandler', 'FileHandler',
'HTTPBasicAuthHandler', 'HTTPCookieProcessor',
'HTTPDefaultErrorHandler',
'HTTPDigestAuthHandler', 'HTTPError',
'HTTPErrorProcessor', 'HTTPHandler',
'HTTPPasswordMgr',
'HTTPPasswordMgrWithDefaultRealm',
'HTTPRedirectHandler', 'HTTPSHandler',
'OpenerDirector', 'ProxyBasicAuthHandler',
'ProxyDigestAuthHandler', 'ProxyHandler',
'Request', 'StringIO', 'URLError',
'UnknownHandler', '__builtins__', '__doc__',
'__file__', '__name__', '__package__',
'__version__', '_cut_port_re', '_opener',
'_parse_proxy', '_safe_gethostbyname',
'addinfourl', 'base64', 'bisect',
'build_opener', 'ftpwrapper', 'getproxies',
'hashlib', 'httplib', 'install_opener',
'localhost', 'mimetools', 'os',
'parse_http_list', 'parse_keqv_list',
'posixpath', 'proxy_bypass', 'quote', 'random',
'randombytes', 're', 'request_host', 'socket',
'splitattr', 'splithost', 'splitpasswd',
'splitport', 'splittag', 'splittype',
'splituser', 'splitvalue', 'sys', 'time',
'toBytes', 'unquote', 'unwrap', 'url2pathname',
'urlopen', 'urlparse', 'warnings']
```

□□□□□□□ urlopen() □ urllib.open() □□□□□□□

Request □

□□□□□□ urllib □ urllib2 □□□□□ urllib2 □□□□□□□□ Request □□□ □□□□□

```
>>> req =
urllib2.Request("http://www.itdiffer.com")
```

□□□ Request □□□□□□□□□□□□□□□□□□ urlopen() □□□□□

```
>>> response = urllib2.urlopen(req)
>>> page = response.read()
>>> print page
```

□□□□□□ `urllib.open("http://www.itdiffer.com")` □□□ □□□□□□□□□□

□□□□□ Request □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ get □□□□□□□□□□□□□□□□□□□□□ post □□□□□□□□□□□□□□ Request □□□

```
import urllib
import urllib2
```

```
url = 'http://www.itdiffer.com/register.py'

values = {'name' : 'qiwsir',
          'location' : 'China',
          'language' : 'Python' }

data = urllib.urlencode(values)     # □□
req = urllib2.Request(url, data)    # □□□□□□□
data □□
response = urllib2.urlopen(req)     #□□□□□□□
the_page = response.read()          #□□□□□□□
```

□□□□□□□□□□□□□□□□□□□□□□□□□□ url □□□□□□□□□□□ post □□□ data □□□□□□□□□□□□□□□□□□□□□ Request □□□□□□□□□□□□□□□□ □□□□□□ post □□□□□□□

□□□□□□□□□□ User-Agent □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□ headers □□□□□□□□

```
user_agent = 'Mozilla/4.0 (compatible; MSIE 5.5;
Windows NT)'
headers = { 'User-Agent' : user_agent }
```

□□□□□□ Request □□□

```
req = urllib2.Request(url, data, headers)
```

□□ urlopen() □□□□□

```
response = urllib2.urlopen(req)
```

□□□□□□□□□□urllib2 □□□□□□□□□□□□□□□:

- □□ HTTP Proxy
- □□ Timeout □
- □□ redirect

- □□ cookie

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ google□

□□□

□□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

xml

xml □□□□□□□□□□□□□□□□□

> “□ XML□□□□□□□□□□ 1998 □ 2 □□□□□□□□□□□□□□□□□□□□□□□□
> □□□
> □ WEB□□□□□□□□□□□□□”
>
> ---□Designing With Web Standards Second Edition□, Jeffrey Zeldman

□□ xml □□□□□□□□□□□□□□□□□□□ w3school □□□□□□□□□□□

- XML □□□□□□□□□EXtensible Markup Language□
- XML □□□□□□□□□□□ HTML
- XML □□□□□□□□□□□□□□□□□
- XML □□□□□□□□□□□□□□□□□□□
- XML □□□□□□□□□□□□
- XML □ W3C □□□□□

□□□□□□□□□□□□□□ xml□□□□□w3school□□□

xml □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ web □□□Python □□□□□□

[illegible] xml [illegible]Python [illegible]

- xml.dom.* [illegible]Document Object Model[illegible] DOM API[illegible] xml [illegible] xml[illegible] xml [illegible]
- xml.sax.* [illegible]simple API for XML[illegible] SAX [illegible] xml [illegible]
- xml.parser.expat[illegible] C [illegible] expat [illegible] expat [illegible] SAX [illegible] expat [illegible]
- xml.etree.ElementTree ([illegible] ET)[illegible] Python [illegible] API[illegible] DOM[illegible]ET [illegible] [illegible] API [illegible] SAX[illegible]ET [illegible] ET.iterparse [illegible] “[illegible]” [illegible]ET [illegible] SAX [illegible] API [illegible]

[illegible] xml.etree.ElementTree

ElementTree [illegible] Python [illegible] xml.etree.ElementTree [illegible] xml.etree.cElementTree [illegible]

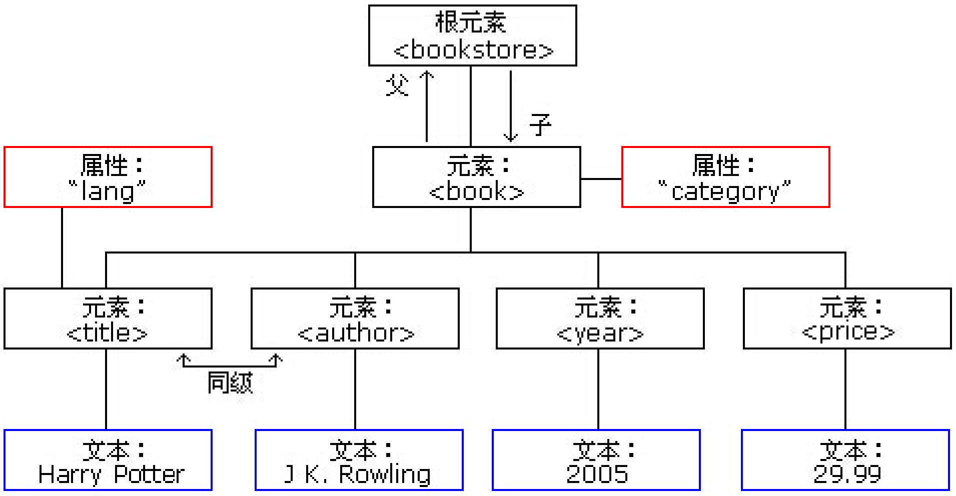
[illegible] Python2.x[illegible]

```
try:
    import xml.etree.cElementTree as ET
except ImportError:
    import xml.etree.ElementTree as ET
```

[illegible] Python3.3 [illegible] `import xml.etree.ElementTree as ET` [illegible] Python3.x [illegible] Python2.x [illegible]

□□□□

□□□□□ xml □□□□□□□□□□□□ w3school □□□□□□□


根元素
<bookstore>
父
子
属性：
"lang"
元素：
<book>
属性：
"category"
元素：
<title>
元素：
<author>
元素：
<year>
元素：
<price>
同级
文本：
Harry Potter
文本：
J K. Rowling
文本：
2005
文本：
29.99

□□□□ xml □□□□□□□□□□□□□□□□□ ET □□□□□□□□□ xml □□□□□

```
<bookstore>
    <book category="COOKING">
        <title lang="en">Everyday
Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="CHILDREN">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
        <book category="WEB">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

[illegible] xml [illegible] 22601.xml [illegible]

```
>>> import xml.etree.cElementTree as ET
```

[illegible] try...except...[illegible]

```
>>> tree = ET.ElementTree(file="22601.xml")
>>> tree
<ElementTree object at 0xb724cc2c>
```

[illegible] xml [illegible]element [illegible]

[illegible] xml [illegible]

```
>>> root = tree.getroot()      #[illegible]
>>> root.tag
'bookstore'
>>> root.attrib
{}
```

[illegible]

```
>>> for child in root:
...     print child.tag, child.attrib
...
book {'category': 'COOKING'}
book {'category': 'CHILDREN'}
book {'category': 'WEB'}
```

[illegible]

```
>>> root[0].tag
'book'
>>> root[0].attrib
```

```
{'category': 'COOKING'}
>>> root[0].text        #□□□
'\n        '
```

□□□□□□□□□□

```
 >>> root[0][0].tag
'title'
>>> root[0][0].attrib
{'lang': 'en'}
>>> root[0][0].text
'Everyday Italian'
```

□□ ElementTree □□□□□□ iter □□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> for ele in tree.iter(tag="book"):        #□□
□□□ book □□□
...     print ele.tag, ele.attrib
...
book {'category': 'COOKING'}
book {'category': 'CHILDREN'}
book {'category': 'WEB'}

>>> for ele in tree.iter(tag="title"):        #□
□□□□ title □□□
...     print ele.tag, ele.attrib, ele.text
...
title {'lang': 'en'} Everyday Italian
title {'lang': 'en'} Harry Potter
title {'lang': 'en'} Learning XML
```

□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> for ele in tree.iter():
...     print ele.tag, ele.attrib
...
bookstore {}
```

```
book {'category': 'COOKING'}
title {'lang': 'en'}
author {}
year {}
price {}
book {'category': 'CHILDREN'}
title {'lang': 'en'}
author {}
year {}
price {}
book {'category': 'WEB'}
title {'lang': 'en'}
author {}
year {}
price {}
```

[illegible] xpath[illegible] xpath [illegible]

```
>>> for ele in tree.iterfind("book/title"):
...     print ele.text
...
Everyday Italian
Harry Potter
Learning XML
```

[illegible] findall() [illegible]

```
>>> for ele in tree.findall("book"):
...     title = ele.find('title').text
...     price = ele.find('price').text
...     lang = ele.find('title').attrib
...     print title, price, lang
...
Everyday Italian 30.00 {'lang': 'en'}
```

```
Harry Potter 29.99 {'lang': 'en'}
Learning XML 39.95 {'lang': 'en'}
```

[illegible]

[illegible] xml [illegible] xml [illegible]

```
>>> root[1].tag
'book'
>>> del root[1]
>>> for ele in root:
...     print ele.tag
...
book
book
```

[illegible] book [illegible]

[illegible] ElementTree [illegible]

```
>>> import os
>>> outpath = os.getcwd()
>>> file = outpath + "/22601.xml"
```

[illegible]

```
>>> tree.write(file)
```

[illegible]

[illegible]

```
>>> for price in root.iter("price"):        #[illegible]
[illegible]
```

```
...     print price.text
...
30.00
39.95
>>> for price in root.iter("price"):        #□□□
□ 7 □□□□□□□□□□
...     new_price = float(price.text) + 7
...     price.text = str(new_price)
...     price.set("updated","up")
...
>>> tree.write(file)
```

□□□□□□

```
<bookstore>
    <book category="COOKING">
        <title lang="en">Everyday
Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price updated="up">37.0</price>
    </book>
    <book category="WEB">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price updated="up">46.95</price>
    </book>
</bookstore>
```

□□□□□□□□□□□ price □□□□□□□□□□□□□□□□□

□□□ `del` □□□□□□□□□□□□□□□□□□□□□□□□□□□ remove() □□□□□□□□□ `price > 40` □□□□□□□□□

```
>>> for book in root.findall("book"):
...     price = book.find("price").text
...     if float(price) > 40.0:
...         root.remove(book)
...
>>> tree.write(file)
```

[illegible]

```
<bookstore>
    <book category="COOKING">
        <title lang="en">Everyday
Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price updated="up">37.0</price>
    </book>
</bookstore>
```

[illegible]

```
>>> import xml.etree.cElementTree as ET
>>> tree = ET.ElementTree(file="22601.xml")
>>> root = tree.getroot()
>>> ET.SubElement(root, "book")        #[illegible] root [illegible]
[illegible] book [illegible]
<Element 'book' at 0xb71c7578>
>>> for ele in root:
...    print ele.tag
...
book
book
>>> b2 = root[1]                       #[illegible]
book [illegible]
>>> b2.text = "Python"                 #[illegible]
>>> tree.write("22601.xml")
```

[illegible]

```
<bookstore>
    <book category="COOKING">
        <title lang="en">Everyday
Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price updated="up">37.0</price>
    </book>
    <book>python</book>
</bookstore>
```

[illegible]

ET [illegible]

Element [illegible]

[illegible]

- tag[illegible]string[illegible]
- text[illegible]string[illegible]
- attrib[illegible]dictionary[illegible]
- tail[illegible]string[illegible]

[illegible]

- clear()[illegible]text [illegible] tail [illegible] None
- get(key, default=None)[illegible] key [illegible] default [illegible]
- items()[illegible](key, value[illegible]
- keys()[illegible]
- set(key, value)[illegible]

□□□□□□□

- append(subelement)□□□□□□□□
- extend(subelements)□□□□□□□□□□□□□□
- find(match)□□□□□□□□□□□□□□□□□□□ tag □ path
- findall(match)□□□□□□□□□□□□□□□□□□ tag □ path
- findtext(match)□□□□□□□□□□□□□□□ text □□□□□□□□□ tag □ path
- insert(index, element)□□□□□□□□□□□
- iter(tag=None)□□□□□□□□□□□□□□□□□ tag □□□□□□□
- iterfind(match)□□□ tag □ path □□□□□□□
- itertext()□□□□□□□□□□ text □
- remove(subelement)□□□□□□

ElementTree □□

- find(match)
- findall(match)
- findtext(match, default=None)
- getroot()□□□□□□.
- iter(tag=None)
- iterfind(match)
- parse(source, parser=None)□□□ xml □□□source □□□□□□□□□□□□□.
- write(file, encoding="us-ascii", xml_declaration=None, default_namespace=None,method="xml")

□□□□

□□□□□□□□□□□□□□□□□□□□□□Python xml □□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□

> □□□□□
>
> □□ ElementTree□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
> □□□□□□□□□□□□□□□.

```
#!/usr/bin/Python
# -*- coding=utf-8 -*-
# author : wklken@yeah.net
# date: 2012-05-25
# version: 0.1

from xml.etree.ElementTree import
ElementTree,Element

def read_xml(in_path):
    '''
        □□□□□ xml □□
        in path: xml □□
        return: ElementTree
    '''
    tree = ElementTree()
    tree.parse(in_path)
    return tree

def write_xml(tree, out_path):
    '''
        □ xml □□□□
        tree: xml □
        out_path: □□□□
    '''
    tree.write(out path, encoding="utf-
8",xml_declaration=True)
```

```
def if_match(node, kv_map):
    '''
        □□□□□□□□□□□□□□□□□□
        node: □□
        kv_map: □□□□□□□□□ map
    '''
    for key in kv_map:
        if node.get(key) != kv_map.get(key):
            return False
    return True

#---------------search -----

def find_nodes(tree, path):
    '''
        □□□□□□□□□□□□□
        tree: xml □
        path: □□□□
    '''
    return tree.findall(path)

def get_node_by_keyvalue(nodelist, kv_map):
    '''
        □□□□□□□□□□□□□□□□□□□□
        nodelist: □□□□
        kv_map: □□□□□□□□ map
    '''
    result_nodes = []
    for node in nodelist:
        if if_match(node, kv_map):
            result_nodes.append(node)
    return result_nodes

#---------------change -----

def change_node_properties(nodelist, kv_map,
is delete=False):
    '''
        □□/□□ /□□ □□□□□□□□□
        nodelist: □□□□
        kv_map:□□□□□□ map
    '''
    for node in nodelist:
        for key in kv map:
            if is delete:
                if key in node.attrib:
                    del node.attrib[key]
            else:
                node.set(key, kv_map.get(key))

def change node text(nodelist, text,
is add=False, is_delete=False):
    '''
        □□/□□/□□□□□□□□□
        nodelist:□□□□
        text : □□□□□□
    '''
    for node in nodelist:
        if is add:
            node.text += text
        elif is delete:
            node.text = ""
        else:
            node.text = text

def create_node(tag, property_map, content):
    '''
        □□□□□□
        tag:□□□□
        property map:□□□□□□ map
        content: □□□□□□□□□□□□
        return □□□
    '''
    element = Element(tag, property_map)
    element.text = content
    return element

def add_child_node(nodelist, element):
    '''
        □□□□□□□□□□
        nodelist: □□□□
        element: □□□
    '''
    for node in nodelist:
        node.append(element)

def del_node_by_tagkeyvalue(nodelist, tag,
kv_map):
    '''
        □□□□□□□□□□□□□□□□□□□
        nodelist: □□□□□
        tag:□□□□□
        kv_map: □□□□□□□□
    '''
    for parent_node in nodelist:
        children = parent_node.getchildren()
        for child in children:
            if child.tag == tag and
if match(child, kv_map):
                parent_node.remove(child)



if __name__ == "__main__":

    #1. □□ xml □□
    tree = read_xml("./test.xml")

    #2. □□□□
    #A. □□□□□
    nodes = find nodes(tree,
"processers/processer")

    #B. □□□□□□□□□□□
    result nodes = get node_by_keyvalue(nodes,
{"name":"BProcesser"})

    #C. □□□□□□
    change_node_properties(result_nodes, {"age":
"1"})

    #D. □□□□□□
    change node properties(result_nodes,
{"value":""}, True)

    #3. □□□□
    #A.□□□□
    a = create node("person",
{"age":"15","money":"200000"}, "this is the
firest content")

    #B.□□□□□□□□
    add_child_node(result_nodes, a)

    #4. □□□□
    #□□□□□
    del parent nodes = find nodes(tree,
"processers/services/service")

    #□□□□□□□□□□□
    target del node =
del node by tagkeyvalue(del parent nodes,
"chain", {"sequency" : "chain1"})

    #5. □□□□□□
    #□□□□
    text nodes =
get node by keyvalue(find nodes(tree,
"processers/services/service/chain"),
{"sequency":"chain3"})
    change_node_text(text_nodes, "new text")

    #6. □□□□□□□
    write_xml(tree, "./out.xml")
```

□□□□□□□ xml □□□□

```
<?xml version="1.0" encoding="UTF-8"?>
<framework>
    <processers>
        <processer name="AProcesser"
file="lib64/A.so"
            path="/tmp">
        </processer>
        <processer name="BProcesser"
file="lib64/B.so" value="fordelete">
        </processer>
        <processer name="BProcesser"
file="lib64/B.so2222222"/>

        <services>
            <service name="search"
prefix="/bin/search?"

output_formatter="OutPutFormatter:service_inc">

                <chain sequency="chain1"/>
                <chain sequency="chain2">
</chain>
            </service>
            <service name="update"
```

```
prefix="/bin/update?">
                <chain sequency="chain3"
value="fordelete"/>
            </service>
        </services>
    </processers>
</framework>
```

□□□□□□□□□□□□□□□

```
 <?xml version='1.0' encoding='utf-8'?>
<framework>
    <processers>
        <processer file="lib64/A.so"
name="AProcesser" path="/tmp">
        </processer>
        <processer age="1" file="lib64/B.so"
name="BProcesser">
            <person age="15" money="200000">this
is the firest content</person>
        </processer>
        <processer age="1"
file="lib64/B.so2222222" name="BProcesser">
            <person age="15" money="200000">this
is the firest content</person>
        </processer>

        <services>
            <service name="search"
output formatter="OutPutFormatter:service_inc"
                prefix="/bin/search?">

                <chain sequency="chain2" />
            </service>
            <service name="update"
prefix="/bin/update?">
```

```
                <chain sequency="chain3"
value="fordelete">new text</chain>
            </service>
        </services>
    </processers>
</framework>
```

[illegible]

[illegible]qiwsir@126.com,**[illegible]**

□□□ (8)

json

□□□□□□□□xml □□□□□□□□□□□□□□□ json□□□□□□□□□□□□□□□□□□□□□□□ web □□□□□□□□□□□□□□□□□□□□□□□ json □□□□□

> JSON□JavaScript Object Notation□□□□□□□□□·□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ JSON □ Javascript □□□□□□□ JSON □□□□□□□□□□□□□□□□□□□□ C □□□□□□□□□□

□□ json □□□□□□□□□□□□□□□□□□http://www.json.org

□□□□□□□□□□□□□□□ json □□□□

> JSON □□□□□□□□

- “□□/□”□□□□□A collection of name/value pairs□□□□□□□□□□□□□□□□□object□□□□□record□□□□□struct□□□□□dictionary□□□□□□□hash table□□□□□□keyed list□□□□□□ □associative array□□
- □□□□□□□An ordered list of values□□□□□□□□□□□□□□□□□□array□□

python □□□□□ json □□□□□□□□□□□□□□□□□□□

- □□□□encoding□□□□ Python □□□□□□□ json □□□
- □□□□□decoding□□ json □□□□□□□□□□ Python □□□□

□□□□

json □□□□ xml □□□□□□

```
>>> import json
>>> json.  all
['dump', 'dumps', 'load', 'loads', 
'JSONDecoder', 'JSONEncoder']
```

encoding: dumps()

```
>>> data = [{"name":"qiwsir", "lang":("python", 
"english"), "age":40}]
>>> print data
[{'lang': ('python', 'english'), 'age': 40, 
'name': 'qiwsir'}]
>>> data json = json.dumps(data)
>>> print data json
[{"lang": ["python", "english"], "age": 40, 
"name": "qiwsir"}]
```

encoding □□□□□□□□□□□□□□□ data □ data_json □□□——lang □□□□□□□□□□□□□□□□

```
>>> type(data_json)
<type 'str'>
>>> type(data)
<type 'list'>
```

□ Python □□□□□ json □□□□□□□□□□□□□□□□□□

Python==>	**json**
dict	object
list, tuple	array

Python==>	**json**
str, unicode	string
int, long, float	number
True	true
False	false
None	null

decoding: loads()

decoding □□□□□□□□□□□□

```
>>> new data = json.loads(data_json)
>>> new data
[{u'lang': [u'python', u'english'], u'age': 40,
u'name': u'qiwsir'}]
```

□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□

json==>	**Python**
object	dict
array	list
string	unicode
number(int)	int, long
number(real)	float

json==>	**Python**
true	True
false	False
null	None

□□□□

□□□ data □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□json □ dumps() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> data_j = json.dumps(data, sort_keys=True, 
indent=2)
>>> print data_j
[
  {
    "age": 40,
    "lang": [
      "python",
      "english"
    ],
    "name": "qiwsir"
  }
]
```

`sort_keys=True` □□□□□□□□□□□□□□`indent=2` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□ **json** □□□

□□ json □□□ `load()` □ `dump()` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> import tempfile    #□□□□□□
>>> data
[{'lang': ('Python', 'english'), 'age': 40,
'name': 'qiwsir'}]
>>> f = tempfile.NamedTemporaryFile(mode='w+')
>>> json.dump(data, f)
>>> f.flush()
>>> print open(f.name, "r").read()
[{"lang": ["Python", "english"], "age": 40,
"name": "qiwsir"}]
```

□□□□□□□

□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□

□□□□□□□Json □□□□ Python □ json □□□□□

```
#!/usr/bin/env Python
# coding=utf-8

import json

class Person(object):
    def   init  (self,name,age):
        self.name = name
        self.age = age

    def   repr  (self):
        return 'Person Object name : %s , age :
%d' % (self.name,self.age)


def object2dict(obj):    #convert Person to dict
    d = {}
    d['__class__'] = obj.__class__.__name__
    d['__module__'] = obj.__module__
    d.update(obj.__dict__)
    return d

def dict2object(d):     #convert dict ot Person
    if '__class__' in d:
        class_name = d.pop('__class__')
        module_name = d.pop('__module__')
        module = __import__(module_name)
        class_ = getattr(module, class_name)
        args = dict((key.encode('ascii'), value)
for key,value in d.items())    #get args
        inst = class_(**args)    #create new
instance
    else:
        inst = d
    return inst



if __name__ == '__main__':
    p = Person('Peter',40)
    print p
    d = object2dict(p)
    print d
    o = dict2object(d)
    print type(o), o

    dump = json.dumps(p, default=object2dict)
    print dump
    load = json.loads(dump,
object_hook=dict2object)
    print load
```

□□□

[illegible]qiwsir@126.com,[illegible]

[illegible]

[illegible] Python [illegible] Python [illegible] master [illegible]·[illegible]

> [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] github.com [illegible]

[illegible] zip [illegible] tar.zip[illegible] tar.bz2 [illegible] setup.py [illegible] Linux [illegible] Mac([illegible] ubuntu[illegible])[illegible] shell[illegible]

```
Python setup.py install
```

[illegible] windows[illegible]

[illegible] Python [illegible],windows [illegible] `C:\Python2.7\Lib\site-`

`packages`[illegible]Linux [illegible] `/usr/local/lib/python2.7/dist-packages`[illegible]Mac [illegible] `/Library/Python/2.7/site-packages`[illegible]

[illegible] site-packages [illegible]

[illegible]pip

[illegible] pip [illegible] easy_install [illegible]https://pypi.Python.org/pypi

[illegible] pip[illegible]Python [illegible] ubuntu [illegible] Linux[illegible] ubuntu [illegible]

Debian and Ubuntu:

```
sudo apt-get install Python-pip
```

Fedora and CentOS:

```
sudo yum install python-pip
```

[illegible]get-pip.py[illegible] `Python get-pip.py` [illegible] windows[illegible]

pip [illegible] `pip install XXXXXX` [illegible]XXXXXX [illegible]

[illegible]

[illegible]requests [illegible]

□ requests □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ urllib □ urllib2 □□□□□□□□□□□□□□□□□□□□□□ requests □□□□□□□□□□□ □ http □□□□ get □ post □□□□□□□□□□□□□“□□□□□”□

□□□□□□□□□□□ 1world0x00 □□□□□□□□□□□□

□□

```
pip install requests
```

□□□□□□□□□□□□□

```
>>> import requests
>>> dir(requests)
['ConnectionError', 'HTTPError', 'NullHandler',
'PreparedRequest', 'Request',
'RequestException', 'Response', 'Session',
'Timeout', 'TooManyRedirects', 'URLRequired',
'  author  ', '  build  ', '  builtins  ',
'  copyright  ', '  doc  ', '  file  ',
'  license  ', '  name  ', '  package  ',
'  path  ', '  title  ', '  version  ',
'adapters', 'api', 'auth', 'certs', 'codes',
'compat', 'cookies', 'delete', 'exceptions',
'get', 'head', 'hooks', 'logging', 'models',
'options', 'packages', 'patch', 'post', 'put',
'request', 'session', 'sessions',
'status_codes', 'structures', 'utils']
```

□□□□□□□□□□□□□ http □□□□□ get□cookies□post □□□□□□□

get □□

```
>>> r = requests.get("http://www.itdiffer.com")
```

□□□□□□□□□□□□□

```
>>> r.cookies
<<class 'requests.cookies.RequestsCookieJar'>[]>
```

□□□□□□□□□□□□□ cookies □□□□□□□□□

```
>>> r =
requests.get("http://www.1world0x00.com")
>>> r.cookies
<<class 'requests.cookies.RequestsCookieJar'>
[Cookie(version=0, name='PHPSESSID',
value='bugi70k7f9rrg51emsvatveda2', port=None,
port specified=False,
domain='www.1world0x00.com',
domain specified=False,
domain initial dot=False, path='/',
path specified=True, secure=False, expires=None,
discard=True, comment=None, comment_url=None,
rest={}, rfc2109=False)]>
```

□□□□□□□□□□□□□□□□□□□□

```
>>> r.headers
{'x-powered-by': 'PHP/5.3.3', 'transfer-
encoding': 'chunked', 'set-cookie':
'PHPSESSID=bugi70k7f9rrg51emsvatveda2; path=/',
'expires': 'Thu, 19 Nov 1981 08:52:00 GMT',
'keep-alive': 'timeout=15, max=500', 'server':
'Apache/2.2.15 (CentOS)', 'connection': 'Keep-
Alive', 'pragma': 'no-cache', 'cache-control':
'no-store, no-cache, must-revalidate, post-
check=0, pre-check=0', 'date': 'Mon, 10 Nov 2014
01:39:03 GMT', 'content-type': 'text/html;
charset=UTF-8', 'x-pingback':
'http://www.1world0x00.com/index.php/action/xmlr
pc'}
```

```
>>> r.encoding
'UTF-8'

>>> r.status_code
200
```

□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> print r.text

<!DOCTYPE html>
<html lang="zh-CN">
  <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>1world0x00sec</title>
  <link rel="stylesheet"
href="http://www.1world0x00.com/usr/themes/defau
lt/style.min.css">
  <link rel="canonical"
href="http://www.1world0x00.com/" />
  <link rel="stylesheet" type="text/css"
href="http://www.1world0x00.com/usr/plugins/Code
Box/css/codebox.css" />
  <meta name="description" content="□□□□□□□□□□□□
□□□□□" />
  <meta name="keywords" content="php" />
  <link rel="pingback"
href="http://www.1world0x00.com/index.php/action
/xmlrpc" />

  ......
```

□□□□□□requests □□□ http □□□□□□□□□□□□□□□□□□□□□ r.text □□□requests □□□□□□□□□□□□□□□□□ requests □□□□□□□□□□□□□□ r.coding □□□□□□□

```
>>> r.content
'\xef\xbb\xbf\xef\xbb\xbf<!DOCTYPE html>\n<html
lang="zh-CN">\n  <head>\n    <meta charset="utf-
8">\n    <meta name="viewport"
content="width=device-width, initial-
scale=1.0">\n    <title>1world0x00sec</title>\n
<link rel="stylesheet"
href="http://www.1world0x00.com/usr/themes/defau
lt/style.min.css">\n            <link ......

□□□□□□□□□□□□□□□□□□
```

post □□

requests □□ post □□□□□□□□□□□□□□□□□□□□□——□□□□□ html □□□□□□□□□□□□□□□□□□□□□□ data □□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> import requests
>>> payload = {"key1":"value1","key2":"value2"}
>>> r = requests.post("http://httpbin.org/post")
>>> r1 =
requests.post("http://httpbin.org/post",
data=payload)
```

http □□

```
>>> r.headers['content-type']
'application/json'
```

[illegible]`'CONTENT-TYPE'`[illegible]

[illegible]

```
>>> r.headers['content-type'] = 'adad'
>>> r.headers['content-type']
'adad'
```

[illegible]

[illegible] requests [illegible]http://requests-docs-cn.readthedocs.org/zh_CN/latest/index.html

---

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

8

[illegible]

[illegible]

[illegible](1)[illegible]

[illegible]

[illegible]

[illegible]

pickle

pickle [illegible] cpickle[illegible] `import pickle` [illegible] `import cpickle as pickle` [illegible]

```
>>> import pickle
>>> integers = [1, 2, 3, 4, 5]
>>> f = open("22901.dat", "wb")
>>> pickle.dump(integers, f)
>>> f.close()
```

[illegible] `pickle.dump(integers, f)` [illegible] integers [illegible] 22901.dat [illegible]

`pickle.dump(obj,file[,protocol])`

- obj[illegible]

- file□□□□□□□□□□□□□□□□□□□□□□□□□□□ write() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ StringIO □□□□□□□□□□□□□□□□□□□□
- protocol□□□□□□□□ False□□□□ 0□□□□ ASCII □□□□□□□□□□□□ 1 □□ True□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□

```
>>> import pickle
>>> d = {}
>>> integers = range(9999)
>>> d["i"] = integers        #□□□□□ dict □□□□□□□
□□

>>> f = open("22902.dat", "wb")
>>> pickle.dump(d, f)           #□□□□ ascii □□□□
□□
>>> f.close()

>>> f = open("22903.dat", "wb")
>>> pickle.dump(d, f, True)     #□□□□□□□□□□□□□
>>> f.close()

>>> import os
>>> s1 = os.stat("22902.dat").st_size    #□□□□□□
□□□
>>> s2 = os.stat("22903.dat").st_size

>>> print "%d, %d, %.2f%%" % (s1, s2,
(s2+0.0)/s1*100)
68903, 29774, 43.21%
```

□□□□□□□□□□□□□□□□□□□□□ ascii □□□□□□□□□□□□□□□□□□□ 43%□

□□□□□□□□□□□□□□□□□□□□□□□□□□ dump() □□□ True □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> integers = pickle.load(open("22901.dat",
"rb"))
>>> print integers
[1, 2, 3, 4, 5]
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> f = open("22903.dat", "rb")
>>> d = pickle.load(f)
>>> print d
{'i': [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,
13, ....   #□□□□□□□}
>>> f.close()
```

□□□

```
>>> import cPickle as pickle        #cPickle □□
>>> import StringIO                 #□□□□□□□□□□□
file □□□□□□□□□□□□□□□“□□”

>>> class Book(object):             #□□□□□□□
...     def   init  (self,name):
...         self.name = name
...     def my book(self):
...         print "my book is: ", self.name
...

>>> pybook = Book("<from beginner to master>")
>>> pybook.my book()
my book is:  <from beginner to master>

>>> file = StringIO.StringIO()
>>> pickle.dump(pybook, file, 1)
>>> print file.getvalue()           #□□“□□”□□□□□
```

```
□□□□□□
ccopy reg
 reconstructor
q(c _main__
Book
qc  builtin__
object
qNtRq}qUnameqU<from beginner to master>sb.

>>> pickle.dump(pybook, file)       #□□□□□□□□□□□
□□□□□□
>>> print file.getvalue()           #□□□□□□□□□□□
□
ccopy reg
 reconstructor
q(c _main__
Book
qc  builtin__
object
qNtRq}qUnameqU<from beginner to
master>sb.ccopy_reg
 reconstructor
p1
(c  main__
Book
p2
c  builtin__
object
p3
NtRp4
(dp5
S'name'
p6
S'<from beginner to master>'
p7
sb.
```

[illegible]

```
>>> file.seek(0)       #[illegible]
>>> pybook2 = pickle.load(file)
>>> pybook2.my_book()
my book is:  <from beginner to master>
>>> file.close()
```

shelve

pickle [illegible]pickle [illegible] shelve[illegible]

shelve [illegible]

```
>>> import shelve
>>> s = shelve.open("22901.db")
>>> s["name"] = "www.itdiffer.com"
>>> s["lang"] = "python"
>>> s["pages"] = 1000
>>> s["contents"] = {"first":"base
knowledge","second":"day day up"}
>>> s.close()
```

[illegible]

```
>>> s = shelve.open("22901.db")
>>> name = s["name"]
>>> print name
www.itdiffer.com
>>> contents = s["contents"]
>>> print contents
{'second': 'day day up', 'first': 'base
knowledge'}
```

□□□□□□□ for □□□□□

```
>>> for k in s:
...     print k, s[k]
...
contents {'second': 'day day up', 'first': 'base knowledge'}
lang python
pages 1000
name www.itdiffer.com
```

□□□□□□□□□□□□□□□□□□□□□□s□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

```
>>> f = shelve.open("22901.db")
>>> f["author"]
['qiwsir']
>>> f["author"].append("Hetz")    #□□□□□□
>>> f["author"]                   #□□□□□
['qiwsir']
>>> f.close()
```

□□

```
>>> f = shelve.open("22901.db", writeback=True)    #□□□□□ True
>>> f["author"].append("Hetz")
>>> f["author"]                   #□□□□
['qiwsir', 'Hetz']
>>> f.close()
```

□□ for □□□□□

```
>>> f = shelve.open("22901.db")
>>> for k,v in f.items():
```

```
...     print k,": ",v
...
contents :  {'second': 'day day up', 'first': 
'base knowledge'}
lang :  python
pages :  1000
author :  ['qiwsir', 'Hetz']
name :  www.itdiffer.com
```

shelve □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

mysql [illegible] (1)

[illegible]——[illegible]“[illegible]”[illegible]

> [illegible]

[illegible]

- [illegible]MySQL[illegible]Microsoft Access[illegible]SQL Server[illegible]Oracle[illegible]...
- [illegible]MongoDB[illegible]BigTable(Google)[illegible]...
- [illegible]Apache Cassandra(Facebook)[illegible]LevelDB(Google) ...

[illegible] MySQL [illegible]

[illegible]

MySQL [illegible]

> MySQL[illegible]/maɪ ˌɛskjuːˈɛl/ "My S-Q-L",[1][illegible]/maɪ ˈsiːkwəl/ "My Sequel"[illegible] MySQL AB [illegible] 2008 [illegible]Sun Microsystems[illegible]2009 [illegible]Oracle[illegible]MySQL [illegible] Oracle [illegible]

> MySQL □□
> Internet □□□□□□□□□□□ MySQL □□□□□□□□□□□□□□□□□□□□□□□
> □□□□□□□Google □ Facebook □□□□□□□□□□□□□□□ LAMP □
> □“M”□□□□ MySQL□
>
> □□□□□□□□□□□Oracle □□□□ MySQL □□□□□□□□□□□□□□□□□□□□
> □□□□□□ OpenSolaris □□□□□□□□□□□□□□□□□ Oracle □□□□□□□
> □ MySQL □□□□MySQL □□□□□□□□□□□□□□□□□□□□□□□ MySQL
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 2013 □□□□□□□ MySQL □□□
> MariaDB □□□□

□□□□□□MySQL □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

□□□□□□□□□□□□ MySQL□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□ Linux □□□□□□□□□□□□□□□□□□□□□□□□□□□

□□ ubuntu □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Linux
□□□□□□□□□□□□□□ ubuntu□□□□□□□□□□□ from beginner to
master□□□□□□□□ master□□□□□□□Linux □□□□□□□□

□□□□□ shell □□□□□□□□

```
sudo apt-get install mysql-server
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□ MySQL

□□□□□□□□

```
service mysqld start
```

□□ mysql □□□□□□□□□□□□□□□□□□□□□

[illegible] MySQL [illegible]“[illegible]”[illegible] root[illegible]

```
$mysql -u root
```

[illegible] -u root [illegible]firehare[illegible] -u root [illegible] mysql [illegible] firehare [illegible]

[illegible] mysql [illegible]>[illegible] mysql [illegible]

[illegible] Mysql [illegible] root [illegible]Mysql [illegible]

```
mysql> GRANT ALL PRIVILEGES ON *.* TO
root@localhost IDENTIFIED BY "123456";
```

[illegible] 123456 [illegible] root [illegible] 8 [illegible]

[illegible]

[illegible]

[illegible]

```
$ mysql -u root -p
Enter password:
```

[illegible]

```
Welcome to the MySQL monitor.  Commands end with
; or \g.
Your MySQL connection id is 373
Server version: 5.5.38-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its
```

```
affiliates. All rights reserved.

Oracle is a registered trademark of Oracle
Corporation and/or its
affiliates. Other names may be trademarks of
their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to
clear the current input statement.

mysql>
```

[illegible]

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information schema |
| carstore           |
| cutvideo           |
| itdiffer           |
| mysql              |
| performance schema |
| test               |
+--------------------+
```

[illegible]

[illegible] phpmyadmin [illegible]

[illegible]

MySQL [illegible] Python [illegible] Python [illegible]——Python-MySQLdb

[illegible] Python-MySQLdb

Python-MySQLdb [illegible]Python [illegible] mysql [illegible]

[illegible]

[illegible]Python-MySQLdb [illegible] MySQL [illegible]

[illegible] Python-mysqldb:https://pypi.Python.org/pypi/MySQL-Python/

[illegible]

ubuntu [illegible]

```
sudo apt-get install build-essential Python-dev
libmysqlclient-dev
sudo apt-get install Python-MySQLdb
```

[illegible] pip [illegible]

```
pip install mysql-Python
```

[illegible] python [illegible]

```
>>> import MySQLdb
```

[illegible]

[illegible] google [illegible]

[illegible]

[illegible]

```
$ mysql -u root -p
Enter password:
```

[illegible]

```
Welcome to the MySQL monitor.  Commands end with
; or \g.
Your MySQL connection id is 373
Server version: 5.5.38-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2014, Oracle and/or its
affiliates. All rights reserved.

Oracle is a registered trademark of Oracle
Corporation and/or its
affiliates. Other names may be trademarks of
their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to
clear the current input statement.

mysql>
```

[illegible]

```
mysql> create database qiwsirtest character set
utf8;
Query OK, 1 row affected (0.00 sec)
```

[illegible]create database qiwsirtest[illegible] character set utf8[illegible] qiwsirtest[illegible] utf-8 [illegible]

[illegible]Query OK, 1 row affected (0.00 sec)[illegible]:qiwsirtest

[illegible] Python [illegible] mysqldb [illegible] qiwsirtest [illegible]

```
>>> import MySQLdb
>>> conn =
MySQLdb.connect(host="localhost",user="root",pas
swd="123123",db="qiwsirtest",port=3306,charset="
utf8")
```

[illegible]

- host:[illegible] mysql [illegible] localhost[illegible] ip [illegible] localhost [illegible]
- user:[illegible]"root",[illegible] root[illegible]
- passwd:[illegible] user [illegible] mysql [illegible]"123123"[illegible]
- db:[illegible] create [illegible]"qiwsirtest"[illegible]
- port:[illegible]mysql [illegible] 3306[illegible] mysql [illegible]
- charset:[illegible] qiwsirtest [illegible] utf-8 [illegible] mysql [illegible]utf-8 [illegible] utf8,[illegible] Python [illegible] utf-8[illegible]

[illegible]connect [illegible] host[illegible]user[illegible]passwd [illegible] host[illegible] user[illegible]passwd[illegible]db ([illegible])[illegible]port [illegible] port=3306 [illegible]

[illegible] db [illegible] port [illegible] 3306 [illegible].

[illegible] connect [illegible]mysqldb [illegible][illegible]

connect(parameters...)

> Constructor for creating a connection to the database. Returns a Connection Object. Parameters are the same as for the MySQL C API. In addition, there are a few additional keywords that correspond to what you would pass mysql_options() before connecting. Note that some parameters must be specified as keyword arguments! The default value for each parameter is NULL or zero, as appropriate. Consult the MySQL documentation for more details. The important parameters are:

- host: name of host to connect to. Default: use the local host via a UNIX socket (where applicable)
- user: user to authenticate as. Default: current effective user.
- passwd: password to authenticate with. Default: no password.
- db: database to use. Default: no default database.
- port: TCP port of MySQL server. Default: standard port (3306).
- unix_socket: location of UNIX socket. Default: use default location or TCP for remote hosts.
- conv: type conversion dictionary. Default: a copy of MySQLdb.converters.conversions
- compress: Enable protocol compression. Default: no compression.

- connect_timeout: Abort if connect is not completed within given number of seconds. Default: no timeout (?)
- named_pipe: Use a named pipe (Windows). Default: don't.
- init_command: Initial command to issue to server upon connection. Default: Nothing.
- read_default_file: MySQL configuration file to read; see the MySQL documentation for mysql_options().
- read_default_group: Default group to read; see the MySQL documentation for mysql_options().
- cursorclass: cursor class that cursor() uses, unless overridden. Default: MySQLdb.cursors.Cursor. This must be a keyword parameter.
- use_unicode: If True, CHAR and VARCHAR and TEXT columns are returned as Unicode strings, using the configured character set. It is best to set the default encoding in the server configuration, or client configuration (read with read_default_file). If you change the character set after connecting (MySQL-4.1 and later), you'll need to put the correct character set name in connection.charset.

If False, text-like columns are returned as normal strings, but you can always write Unicode strings.

This must be a keyword parameter.

- charset: If present, the connection character set will be changed to this character set, if they are not equal. Support for changing the character set requires MySQL-4.1 and later server; if the server is too old, UnsupportedError will be raised. This option implies use_unicode=True, but you can override this with use_unicode=False, though you probably shouldn't.

If not present, the default character set is used.

This must be a keyword parameter.

- sql_mode: If present, the session SQL mode will be set to the given string. For more information on sql_mode, see the MySQL documentation. Only available for 4.1 and newer servers.

If not present, the session SQL mode will be unchanged.

This must be a keyword parameter.

- ssl: This parameter takes a dictionary or mapping, where the keys are parameter names used by the mysql_ssl_set MySQL C API call. If this is set, it initiates an SSL connection to the server; if there is no SSL support in the client, an exception is raised. This must be a keyword parameter.

□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

MySQL [illegible] (2)

[illegible] qiwsirtest [illegible]“[illegible]”[illegible]

> [illegible] authors [illegible]“[illegible]”[illegible]
>
> [illegible]

[illegible] qiwsirtest [illegible]

username	password	email
qiwsir	123123	qiwsir@gmail.com

[illegible]

[illegible]

[illegible]

```
mysql> use qiwsirtest;
Database changed
```

```
mysql> show tables;
Empty set (0.00 sec)
```

[illegible] `show tables` [illegible]

[illegible]

```
mysql> create table users(id int(2) not null
primary key auto increment,username
varchar(40),password text,email text)default
charset=utf8;
Query OK, 0 rows affected (0.12 sec)
```

[illegible]users[illegible]

```
mysql> show tables;
+----------------------+
| Tables in qiwsirtest |
+----------------------+
| users                |
+----------------------+
1 row in set (0.00 sec)
```

[illegible] qiwsirtest [illegible]users[illegible]

```
mysql> desc users;
+----------+-------------+------+-----+---------
+----------------+
| Field    | Type        | Null | Key | Default
| Extra          |
+----------+-------------+------+-----+---------
+----------------+
| id       | int(2)      | NO   | PRI | NULL
| auto increment |
| username | varchar(40) | YES  |     | NULL
|                |
| password | text        | YES  |     | NULL
|                 |
| email    | text        | YES  |     | NULL
|                 |
+----------+------------+------+-----+---------
+----------------+
4 rows in set (0.00 sec)
```

□□□ users □□□□

□□□

□□□□□□□□□□□□□□□□□□

```
 mysql> select * from users;
Empty set (0.01 sec)
```

□□□□□□□□□□□□□□□□□

```
 mysql> insert into
users(username,password,email)
values("qiwsir","123123","qiwsir@gmail.com");
Query OK, 1 row affected (0.05 sec)

mysql> select * from users;
+----+----------+----------+------------------+
| id | username | password | email            |
+----+----------+----------+------------------+
|  1 | qiwsir   | 123123   | qiwsir@gmail.com |
+----+----------+----------+------------------+
1 row in set (0.00 sec)
```

□□□□□□□□□□□□□□□□□

Python □□□□□

[illegible]

```
>>> import MySQLdb
>>> conn = MySQLdb.connect(host="localhost",user="root",passwd="123123",db="qiwsirtest",charset="utf8")
```

Python [illegible] `MySQLdb.connect()` [illegible]Python [illegible]

- commit()[illegible]
- rollback()[illegible]
- cursor([cursorclass])[illegible] SQL [illegible]
- close()[illegible]

Python [illegible] Python [illegible] SQL [illegible]Python [illegible] SQL [illegible]

```
>>> cur = conn.cursor()
```

[illegible]

[illegible]	[illegible]
close()	[illegible]
execute(query[,args])	[illegible] SQL [illegible]
executemany(query, pseq)	[illegible] pseq [illegible] sql [illegible]
fetchone()	[illegible]

[illegible]	[illegible]
fetchall()	[illegible]
fetchmany([size])	[illegible] size [illegible]
nextset()	[illegible]
scroll(value,mode='relative')	[illegible] mode='relative',[illegible] value [illegible],[illegible] mode='absolute',[illegible] value [illegible].

[illegible]

[illegible] users [illegible]:username="Python",password="123456",email="Python@gmail.com"[illegible]

```
>>> cur.execute("insert into users
(username,password,email) values (%s,%s,%s)",
("Python","123456","Python@gmail.com"))
1L
```

[illegible]"1L"[illegible]n [illegible]"mysql>"[illegible]

```
mysql> select * from users;
+----+----------+----------+------------------+
| id | username | password | email            |
+----+----------+----------+------------------+
|  1 | qiwsir   | 123123   | qiwsir@gmail.com |
+----+----------+----------+------------------+
1 row in set (0.00 sec)
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□"cur.execute()"□□□"MySQLdb.connect"□□□□□□commit()□□□□□□□□□□□□□□□"cur.execute()"□□□□□□□□□□□□□□□

```
>>> conn.commit()
```

□□"mysql>"□□□"select * from users"□□□□

```
mysql> select * from users;
+----+----------+----------+------------------+
| id | username | password | email            |
+----+----------+----------+------------------+
|  1 | qiwsir   | 123123   | qiwsir@gmail.com |
|  2 | python   | 123456   | python@gmail.com |
+----+----------+----------+------------------+
2 rows in set (0.00 sec)
```

□□"CTRL-S"□□□□□□□□□□□ Python □□□□□□□□□□□"execute()"□□□□ sql □□□□□□□□□□□□□□□□□□□□□□□□□□"commit()"□□□

□□□□□□□□□□□□□□"executemany(query,args)".

```
>>> cur.executemany("insert into users
(username,password,email) values (%s,%s,%s)",
(("google","111222","g@gmail.com"),
("facebook","222333","f@face.book"),
("github","333444","git@hub.com"),
("docker","444555","doc@ker.com")))
4L
>>> conn.commit()
```

[illegible]"mysql>"[illegible]

```
mysql> select * from users;
+----+----------+----------+------------------+
| id | username | password | email            |
+----+----------+----------+------------------+
|  1 | qiwsir   | 123123   | qiwsir@gmail.com |
|  2 | python   | 123456   | python@gmail.com |
|  3 | google   | 111222   | g@gmail.com      |
|  4 | facebook | 222333   | f@face.book      |
|  5 | github   | 333444   | git@hub.com      |
|  6 | docker   | 444555   | doc@ker.com      |
+----+----------+----------+------------------+
6 rows in set (0.00 sec)
```

[illegible]"executemany(query, pseq)"[illegible]query [illegible] sql [illegible] pseq [illegible] tuple[illegible] tuple [illegible] tuple[illegible] tuple [illegible] sql [illegible]

[illegible]

[illegible]

[illegible]

```
>>> cur.execute("select * from users")
7L
```

[illegible] users [illegible] 7 [illegible] 7 [illegible]'mysql>'[illegible] 7 [illegible] Python [illegible]

[illegible] fetchall()[illegible]fetchmany(size=None)[illegible]fetchone()[illegible] scroll(value, mode='relative')[illegible]

```
>>> cur.execute("select * from users")
7L
>>> lines = cur.fetchall()
```

□□□□□□□□□□□□□□□□□□ lines □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> for line in lines:
...     print line
...
(1L, u'qiwsir', u'123123', u'qiwsir@gmail.com')
(2L, u'python', u'123456', u'python@gmail.com')
(3L, u'google', u'111222', u'g@gmail.com')
(4L, u'facebook', u'222333', u'f@face.book')
(5L, u'github', u'333444', u'git@hub.com')
(6L, u'docker', u'444555', u'doc@ker.com')
(7L, u'\u8001\u9f50', u'9988',
u'qiwsir@gmail.com')
```

□□□□□□□□□□□□□□□□□□□□□□□□ u'\u8001\u95f5',□□□□□□□□□□□□□ shell □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□

```
>>> cur.execute("select * from users where
id=1")
1L
>>> line first = cur.fetchone()     #□□□□□
>>> print line first
(1L, u'qiwsir', u'123123', u'qiwsir@gmail.com')
```

□□□□□□□□□□□□□□□□□□

```
>>> cur.execute("select * from users")
7L
>>> print cur.fetchall()
((1L, u'qiwsir', u'123123',
u'qiwsir@gmail.com'), (2L, u'python', u'123456',
u'python@gmail.com'), (3L, u'google', u'111222',
u'g@gmail.com'), (4L, u'facebook', u'222333',
u'f@face.book'), (5L, u'github', u'333444',
u'git@hub.com'), (6L, u'docker', u'444555',
u'doc@ker.com'), (7L, u'\u8001\u9f50', u'9988',
u'qiwsir@gmail.com'))
```

□□□□□ cur.execute() □□□□□□□□□□□□□□□□“□□□□□ cur □□□□□□□□□□□□”□□□□□□□□□□□□□□□□□□□ cur.fetchall()□□□□ fechone □□ tuple □□□□□□□□□□□□□□□□□□□□□ tuple□□□□□□ for □□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□

```
>>> print cur.fetchall()
()
```

□□

□□□□□□

□□□ print cur.fetchall() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ print □□□

□□□□□□□□□□□□□□□□□□

```
>>> cur.execute('select * from users')
7L
>>> print cur.fetchone()
(1L, u'qiwsir', u'123123', u'qiwsir@gmail.com')
>>> print cur.fetchone()
(2L, u'python', u'123456', u'python@gmail.com')
```

```
>>> print cur.fetchone()
(3L, u'google', u'111222', u'g@gmail.com')
```

□□

□□□ scroll()

```
 >>> cur.scroll(1)
>>> print cur.fetchone()
(5L, u'github', u'333444', u'git@hub.com')
>>> cur.scroll(-2)
>>> print cur.fetchone()
(4L, u'facebook', u'222333', u'f@face.book')
```

□□□

cur.scroll(n)□□□□cur.scroll(n,"relative")□□□□□□□□□□□□□□□□□□□n □□□□□□□□□□□□□n □□□□□□□□□□□□

□□□□□□□□□□□“□□”□□□□□“□□”□□□□□□□□□□"absolute"

□□□□□□□□□□□□ Python □□□□□□□□□□□□ 0 □□□□

```
 >>> cur.scroll(2,"absolute")    #□□□□□ 2,□□□□□□
>>> print cur.fetchone()        #□□□□□□
(3L, u'google', u'111222', u'g@gmail.com')

>>> cur.scroll(1,"absolute")
>>> print cur.fetchone()
(2L, u'python', u'123456', u'python@gmail.com')

>>> cur.scroll(0,"absolute")    #□□□□□ 0,□□□
tuple □□□□
>>> print cur.fetchone()
(1L, u'qiwsir', u'123123', u'qiwsir@gmail.com')
```

□□□□□□□□ cur.fetchall() □ cur.fetchone() □□ cur.scroll() □□□
□□□□□□□□□□□□□□□□□□□□□□□□□□ 1 □□□□□□□ tuple □□□□

```
>>> cur.fetchmany(3)
((2L, u'Python', u'123456',
u'python@gmail.com'), (3L, u'google', u'111222',
u'g@gmail.com'), (4L, u'facebook', u'222333',
u'f@face.book'))
```

□□□□□□□□□□□□□□□□□□□□□□ tuple □□□□ 1 □□□□□□□□□□□□□□□
□□□□□□□□□ 3 □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□Python □□
□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> cur =
conn.cursor(cursorclass=MySQLdb.cursors.DictCurs
or)
>>> cur.execute("select * from users")
7L
>>> cur.fetchall()
({'username': u'qiwsir', 'password': u'123123',
'id': 1L, 'email': u'qiwsir@gmail.com'},
{'username': u'mypython', 'password': u'123456',
'id': 2L, 'email': u'python@gmail.com'},
{'username': u'google', 'password': u'111222',
'id': 3L, 'email': u'g@gmail.com'}, {'username':
u'facebook', 'password': u'222333', 'id': 4L,
'email': u'f@face.book'}, {'username':
u'github', 'password': u'333444', 'id': 5L,
'email': u'git@hub.com'}, {'username':
u'docker', 'password': u'444555', 'id': 6L,
'email': u'doc@ker.com'}, {'username':
```

```
u'\u8001\u9f50', 'password': u'9988', 'id': 7L,
'email': u'qiwsir@gmail.com'})
```

□□□□□□□□□□□□□□□□□□□□

```
>>> cur.scroll(0,"absolute")
>>> for line in cur.fetchall():
...     print line["username"]
...
qiwsir
mypython
google
facebook
github
docker
□□
```

□□□□□□□□□□□□□“□-□”□

□□□□

□□ commit() □□□□□□

```
>>> cur.execute("update users set username=%s
where id=2",("mypython"))
1L
>>> cur.execute("select * from users where
id=2")
1L
>>> cur.fetchone()
(2L, u'mypython', u'123456',
u'python@gmail.com')
```

[illegible] myPython [illegible] update [illegible]

[illegible]

```
>>> conn.commit()
```

[illegible]

[illegible]

```
>>> cur.close()
>>> conn.close()
```

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

mongodb [illegible] (1)

MongoDB [illegible] Python [illegible] mongodb[illegible] mysql [illegible] mongodb [illegible]

mongodb [illegible] NoSql [illegible]

NoSql[illegible] Not Only Sql,[illegible] web [illegible] API[illegible]

MongoDB [illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

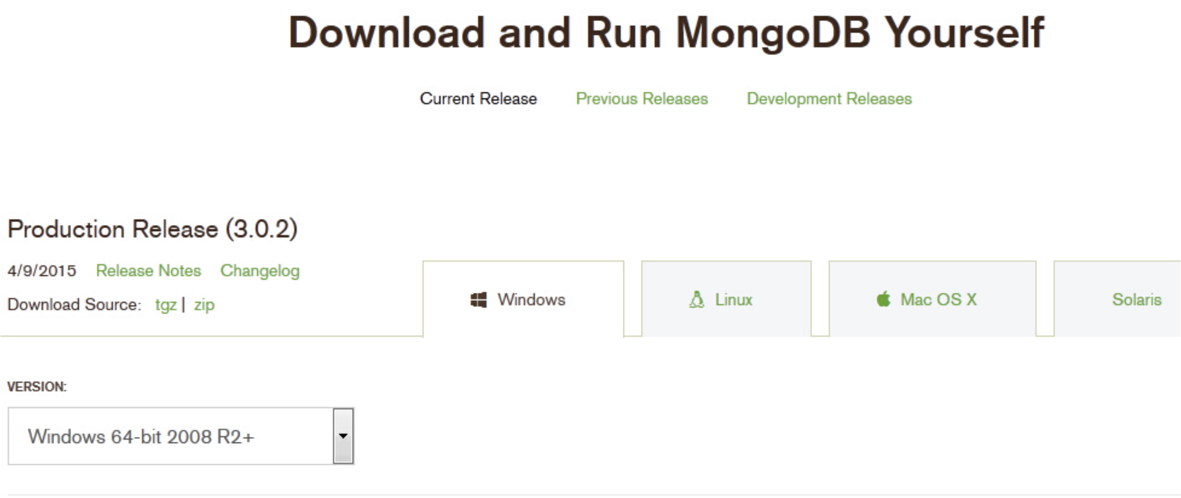
[illegible] mongodb

[illegible] ubuntu [illegible]

```
sudo apt-key adv --keyserver
hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
echo 'deb http://downloads-
distro.mongodb.org/repo/ubuntu-upstart dist
10gen' | sudo tee
/etc/apt/sources.list.d/mongodb.list
sudo apt-get update
sudo apt-get install mongodb-10gen
```

[illegible]Install MongoDB

[illegible]
http://www.mongodb.org/downloads[illegible]



Download and Run MongoDB Yourself

Current Release
Previous Releases
Development Releases

Production Release (3.0.2)

4/9/2015 Release Notes Changelog

Download Source: tgz | zip

Windows
Linux
Mac OS X
Solaris

VERSION: Windows 64-bit 2008 R2+

> [illegible]
>
> window [illegible] MongoDB
>
> NoSQL [illegible]MongoDB[illegible]

> MongoDB [illegible](Ubuntu)
>
> [illegible] Ubuntu [illegible] MongoDB
>
> [illegible] Ubuntu [illegible] MongoDB [illegible]

[illegible] mongodb

[illegible] mongodb [illegible] Python [illegible]

[illegible] `mongo` [illegible] shell[illegible] `>` [illegible] mysql [illegible] shell [illegible]

[illegible] shell [illegible] db[illegible] db[illegible]

```
> use mydb
switched to db mydb
> db
mydb
```

[illegible]

```
> show dbs;
local    0.03125GB
```

[illegible]mongodb [illegible] Python [illegible]

```
> book = {"title":"from beginner to master", "author":"qiwsir", "lang":"python"}
{
    "title" : "from beginner to master",
    "author" : "qiwsir",
    "lang" : "python"
}
> db.books.insert(book)
> db.books.find()
{ " id" : ObjectId("554f0e3cf579bc0767db9edf"), "title" : "from beginner to master", "author" : "qiwsir", "lang" : "Python" }
```

db [illegible] mydb[illegible]books [illegible] mysql [illegible] books [illegible] mysql [illegible]“[illegible]”[illegible] mongodb [illegible]

[illegible] create [illegible] `use xxx` [illegible] `db.xxx` [illegible]“[illegible]”[illegible]

```
> show dbs
local    0.03125GB
mydb    0.0625GB
```

[illegible] mydb [illegible]

[illegible] mongodb [illegible] shell [illegible]“[illegible]”[illegible] Python [illegible]

[illegible] Pymongo

[illegible] Python [illegible] mongodb[illegible] Pymongo[illegible] mysql [illegible]

```
$ sudo pip install Pymongo
```

[illegible]

```
Successfully installed Pymongo
Cleaning up...
```

[illegible]

```
>>> import Pymongo
>>> pymongo.version
'3.0.1'
```

[illegible] 2.8 [illegible]

```
$ sudo pip install Pymongo==2.8
```

[illegible]

[illegible] Python [illegible]

```
>>> import Pymongo
```

[illegible]

[illegible] mongodb

[illegible] Python [illegible] mongdb [illegible] Pymongo [illegible]

```
>>> pymongo.Connection("localhost",27017)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'module' object has no attribute
'Connection'
```

[illegible]

□□□□□□□□□□□□□□ Pymongo□□□ 2.8□□

□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> dir(pymongo)
['ALL', 'ASCENDING', 'CursorType', 'DESCENDING',
'DeleteMany', 'DeleteOne', 'GEO2D',
'GEOHAYSTACK', 'GEOSPHERE', 'HASHED',
'IndexModel', 'InsertOne',
'MAX SUPPORTED WIRE VERSION',
'MIN SUPPORTED WIRE VERSION', 'MongoClient',
'MongoReplicaSetClient', 'OFF',
'ReadPreference', 'ReplaceOne',
'ReturnDocument', 'SLOW ONLY', 'TEXT',
'UpdateMany', 'UpdateOne', 'WriteConcern',
'  builtins  ', '  doc  ', '  file  ',
'  name  ', '  package  ', '  path  ',
' cmessage', 'auth', 'bulk', 'client options',
'collection', 'command cursor', 'common',
'cursor', 'cursor manager', 'database',
'errors', 'get version string', 'has_c',
'helpers', 'ismaster', 'message',
'mongo client', 'mongo replica set client',
'monitor', 'monotonic', 'network', 'operations',
'periodic executor', 'pool', 'read_preferences',
'response', 'results', 'server',
'server description', 'server selectors',
'server type', 'settings', 'son manipulator',
'ssl context', 'ssl support', 'thread_util',
'topology', 'topology description',
'uri parser', 'version', 'version_tuple',
'write_concern']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Connection() □□□□□□□□□□□□□ Pymongo □□□“□□□”□

□□□□□□□□□MongoClient()

```
>>> client = pymongo.MongoClient("localhost",
27017)
```

□□□Python □□□ mongodb □□□□□□

□□□□□□□□□□□□ mydb□□□□□□□□□□□□□□ books□□□□

```
>>> db = client.mydb
```

□□

```
>>> db = client['mydb']
```

□□□□□ mydb□□□□□□□ db□□□□□□□ mongodb □ shell □□□□ db□□□□ db □□ Python □□□□□□□□□□

```
>>> db.collection names()
[u'system.indexes', u'books']
```

□□□□□□□□□□□□□□□□□□ books□□□□□□□□□□□□□□□□□□□ books□

```
>>> books = db["books"]
```

□□

```
>>> books = db.books
```

□□□□□□□□□□□□□□□□□□□□□

□□

□□□ books □□□□□□□ mongodb □□□□□□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□

```
 >>> type(books)
<class 'pymongo.collection.Collection'>

>>> dir(books)
[' BaseObject  codec options',
' BaseObject  read preference',
' BaseObject  write concern',
' Collection  create',
' Collection  create index',
' Collection  database',
' Collection  find and modify',
' Collection  full name', ' Collection  name',
'  call  ', '  class  ', '  delattr  ',
'  dict  ', '  doc  ', '  eq  ', ' _format__',
'  getattr  ', '  getattribute  ',
'  getitem  ', '  hash  ', '  init  ',
'  iter  ', '  module  ', '  ne  ', '  new _',
'  next  ', '  reduce  ', '  reduce ex  ',
'  repr  ', '  setattr  ', '  sizeof  ',
'  str  ', '  subclasshook  ', '  weakref  ',
' command', ' count', ' delete', '_insert',
' socket for primary reads',
' socket for reads', ' socket for writes',
' update', 'aggregate', 'bulk write',
'codec options', 'count', 'create index',
'create indexes', 'database', 'delete many',
'delete one', 'distinct', 'drop', 'drop index',
'drop indexes', 'ensure index', 'find',
'find and modify', 'find one',
'find one and delete', 'find one and replace',
'find one and update', 'full_name', 'group',
'index information',
'initialize ordered bulk op',
'initialize unordered bulk op',
'inline map reduce', 'insert', 'insert many',
'insert one', 'list indexes', 'map reduce',
'name', 'next', 'options', 'parallel scan',
'read preference', 'reindex', 'remove',
'rename', 'replace one', 'save', 'update',
'update many', 'update_one', 'with_options',
'write_concern']
```

□□□□□□□□□□□□□□□□“□□□□”□□□□□□□□□□□□□□□□□ help() □□□□□□□□□□□□□□

```
>>> books.find one()
{u'lang': u'Python', u' id':
ObjectId('554f0e3cf579bc0767db9edf'), u'author':
u'qiwsir', u'title': u'from beginner to master'}
```

□□□□□□□□□□□□□□□ mongodb □ shell □□□□□□ Pymongo □□□□□□□□□□□□□□□□□ mongodb □ shell □□□□□□□

```
> db.books.findOne()
{
    " id" :
ObjectId("554f0e3cf579bc0767db9edf"),
    "title" : "from beginner to master",
    "author" : "qiwsir",
    "lang" : "python"
}
```

□□□□□□

□□□□□ books □□□□□□□□□□□□□□□□□□□□□

□□□□□

```
>>> b2 = {"title":"physics", "author":"Newton",
"lang":"english"}
>>> books.insert(b2)
ObjectId('554f28f465db941152e6df8b')
```

□□
□□□

```
>>> books.find().count()
2
```

□□□□□□□□□□□□□□□□□□□□□ 2□□□□□□□□□□□□□□□□□□

```
>>> books.find one()
{u'lang': u'python', u' id':
ObjectId('554f0e3cf579bc0767db9edf'), u'author':
u'qiwsir', u'title': u'from beginner to master'}
```

□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> for i in books.find():
...     print i
...
{u'lang': u'Python', u' id':
ObjectId('554f0e3cf579bc0767db9edf'), u'author':
u'qiwsir', u'title': u'from beginner to master'}
{u'lang': u'english', u'title': u'physics',
u' id': ObjectId('554f28f465db941152e6df8b'),
u'author': u'Newton'}
```

□ books □□□□□□□ find() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□

```
>>> books.insert({"name":"Hertz"})
ObjectId('554f2b4565db941152e6df8c')
>>> for i in books.find():
...     print i
...
{u'lang': u'Python', u' id':
ObjectId('554f0e3cf579bc0767db9edf'), u'author':
u'qiwsir', u'title': u'from beginner to master'}
```

```
{u'lang': u'english', u'title': u'physics',
u'_id': ObjectId('554f28f465db941152e6df8b'),
u'author': u'Newton'}
{u'_id': ObjectId('554f2b4565db941152e6df8c'),
u'name': u'Hertz'}
```

□□□□□□□□□□□□□□□□□□□□□ mysql □□

```
>>> n1 = {"title":"java", "name":"Bush"}
>>> n2 = {"title":"fortran", "name":"John Warner
Backus"}
>>> n3 = {"title":"lisp", "name":"John
McCarthy"}
>>> n = [n1, n2, n3]
>>> n
[{'name': 'Bush', 'title': 'java'}, {'name':
'John Warner Backus', 'title': 'fortran'},
{'name': 'John McCarthy', 'title': 'lisp'}]
>>> books.insert(n)
[ObjectId('554f30be65db941152e6df8d'),
ObjectId('554f30be65db941152e6df8e'),
ObjectId('554f30be65db941152e6df8f')]
```

□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> books.find().count()
6
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 20 □□□□□□□□□□
16MB□□
□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□`'name'='Bush'`□□□□

```
>>> books.find_one({"name":"Bush"})
{u'_id': ObjectId('554f30be65db941152e6df8d'),
u'name': u'Bush', u'title': u'java'}
```

□□□□□□□□□□□□□□□□

```
>>> for i in books.find().sort("title", pymongo.ASCENDING):
...     print i
...
{u'_id': ObjectId('554f2b4565db941152e6df8c'), u'name': u'Hertz'}
{u'_id': ObjectId('554f30be65db941152e6df8e'), u'name': u'John Warner Backus', u'title': u'fortran'}
{u'lang': u'python', u'_id': ObjectId('554f0e3cf579bc0767db9edf'), u'author': u'qiwsir', u'title': u'from beginner to master'}
{u'_id': ObjectId('554f30be65db941152e6df8d'), u'name': u'Bush', u'title': u'java'}
{u'_id': ObjectId('554f30be65db941152e6df8f'), u'name': u'John McCarthy', u'title': u'lisp'}
{u'lang': u'english', u'title': u'physics', u'_id': ObjectId('554f28f465db941152e6df8b'), u'author': u'Newton'}
```

□□□□"title"□□□□□□□□□□□□ sort() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `Pymongo.DESCEDING` □□□□□□□□□□□□□

```
>>> for i in books.find().sort([("name",pymongo.ASCENDING),("name",pymongo.DESCENDING)]):
...     print i
...
{u'_id': ObjectId('554f30be65db941152e6df8e'), u'name': u'John Warner Backus', u'title': u'fortran'}
{u'_id': ObjectId('554f30be65db941152e6df8f'),
u'name': u'John McCarthy', u'title': u'lisp'}
{u'_id': ObjectId('554f2b4565db941152e6df8c'),
u'name': u'Hertz'}
{u'_id': ObjectId('554f30be65db941152e6df8d'),
u'name': u'Bush', u'title': u'java'}
{u'lang': u'python', u'_id':
ObjectId('554f0e3cf579bc0767db9edf'), u'author':
u'qiwsir', u'title': u'from beginner to master'}
{u'lang': u'english', u'title': u'physics',
u'_id': ObjectId('554f28f465db941152e6df8b'),
u'author': u'Newton'}
```

[illegible] mongodb [illegible]“[illegible]”[illegible] Python [illegible] json [illegible](8)[illegible] Python [illegible] json [illegible] json.dumps() [illegible] `'_id'` [illegible] `del doc['_id']` [illegible] json.dumps() [illegible] json_util [illegible]“Tools for using Python’s json module with BSON documents”[illegible] http://api.mongodb.org/Python/current/api/bson/json_util.html[illegible]

[illegible]

[illegible] name [illegible] Hertz [illegible]

```
>>> books.update({"name":"Hertz"}, {"$set":
{"title":"new physics", "author":"Hertz"}})
{u'updatedExisting': True, u'connectionId': 4,
u'ok': 1.0, u'err': None, u'n': 1}
>>> books.find_one({"author":"Hertz"})
{u'title': u'new physics', u'_id':
ObjectId('554f2b4565db941152e6df8c'), u'name':
u'Hertz', u'author': u'Hertz'}
```

[illegible] `$set` [illegible]

[illegible]

[illegible]	[illegible]
$set	[illegible]
$unset	[illegible]
$inc	[illegible]
$push	[illegible] key [illegible]

[illegible]

[illegible] remove() [illegible]

```
>>> books.remove({"name":"Bush"})
{u'connectionId': 4, u'ok': 1.0, u'err': None,
u'n': 1}
>>> books.find_one({"name":"Bush"})
>>>
```

[illegible] mongodb [illegible]

[illegible]

[illegible]

```
>>> books.create index([("title",
pymongo.DESCENDING),])
u'title_-1'
```

[illegible] Pymongo [illegible] mongodb [illegible]

[illegible]

[illegible]qiwsir@126.com,[illegible]

SQLite [illegible]

SQLite [illegible] MySQL [illegible] MongoDB[illegible]“[illegible]”[illegible] SQLite [illegible] Python [illegible] Python[illegible] SQLite [illegible]SQLite [illegible]

SQLite [illegible] SQL [illegible]

[illegible] mysql [illegible] SQLite [illegible]

- [illegible]
- [illegible]
- [illegible] sql [illegible]

[illegible]

[illegible] SQLite [illegible] Python [illegible]

```
>>> import sqlite3
>>> conn = sqlite3.connect("23301.db")
```

[illegible] mysql [illegible] `sqlite3.connect("23301.db")` [illegible]

[illegible]

```
/2code$ ls 23301.db
23301.db
```

□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> dir(conn)
['DataError', 'DatabaseError', 'Error',
'IntegrityError', 'InterfaceError',
'InternalError', 'NotSupportedError',
'OperationalError', 'ProgrammingError',
'Warning', '__call__', '__class__',
'__delattr__', '__doc__', '__enter__',
'__exit__', '__format__', '__getattribute__',
'__hash__', '__init__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__',
'close', 'commit', 'create_aggregate',
'create_collation', 'create_function', 'cursor',
'enable_load_extension', 'execute',
'executemany', 'executescript', 'interrupt',
'isolation_level', 'iterdump', 'load_extension',
'rollback', 'row_factory', 'set_authorizer',
'set_progress_handler', 'text_factory',
'total_changes']
```

□□□□

□□□ mysql □□□□□□□□□□□□□

```
>>> cur = conn.cursor()
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> dir(cur)
['  class  ', '  delattr  ', '  doc  ',
'  format  ', '  getattribute  ', '  hash  ',
'  init  ', '  iter  ', '  new  ', '  reduce__',
'  reduce ex  ', '  repr  ', '  setattr  ',
'  sizeof  ', '  str  ', '  subclasshook__',
'arraysize', 'close', 'connection',
'description', 'execute', 'executemany',
'executescript', 'fetchall', 'fetchmany',
'fetchone', 'lastrowid', 'next', 'row factory',
'rowcount', 'setinputsizes', 'setoutputsize']
```

□□□□□□□□□□□□`close(), execute(), executemany(), fetchall()`

□□□□□□

□ mysql □□□□□□□□□□ mysql □ shell □□□□□□□□□□□□□□□ sql □□□□ Python □□□□□□□□□□□ sqlite □□□□□□□□□□□□

```
>>> create table = "create table books (title
text, author text, lang text) "
>>> cur.execute(create table)
<sqlite3.Cursor object at 0xb73ed5a0>
```

□□□□□□□ 23301.db □□□□□□□ books□□□□□□□□□□□□□

```
>>> cur.execute('insert into books values ("from
beginner to master", "laoqi", "python")')
<sqlite3.Cursor object at 0xb73ed5a0>
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> conn.commit()
>>> cur.close()
>>> conn.close()
```

[illegible] books[illegible]

[illegible]

[illegible]

[illegible]

```
>>> conn = sqlite3.connect("23301.db")
>>> cur = conn.cursor()
>>> cur.execute('select * from books')
<sqlite3.Cursor object at 0xb73edea0>
>>> print cur.fetchall()
[(u'from beginner to master', u'laoqi',
u'python')]
```

[illegible]

[illegible]

```
>>> books = [("first book","first","c"),
("second book","second","c"), ("third
book","second","python")]
```

[illegible]

```
>>> cur.executemany('insert into books values
(?,?,?)', books)
<sqlite3.Cursor object at 0xb73edea0>
>>> conn.commit()
```

[illegible]

```
>>> rows = cur.execute('select * from books')
>>> for row in rows:
...     print row
...
(u'from beginner to master', u'laoqi',
u'python')
(u'first book', u'first', u'c')
(u'second book', u'second', u'c')
(u'third book', u'second', u'python')
```

□□

□□□□□□□□ cur.execute() □□□□□□ SQL □□□□□□□□□□

```
>>> cur.execute("update books set
title='physics' where author='first'")
<sqlite3.Cursor object at 0xb73edea0>
>>> conn.commit()
```

□□□□□□□□□□□

```
>>> cur.execute("select * from books where
author='first'")
<sqlite3.Cursor object at 0xb73edea0>
>>> cur.fetchone()
(u'physics', u'first', u'c')
```

□□

□ sql □□□□□□□□□□□

```
>>> cur.execute("delete from books where
author='second'")
<sqlite3.Cursor object at 0xb73edea0>
```

```
>>> conn.commit()

>>> cur.execute("select * from books")
<sqlite3.Cursor object at 0xb73edea0>
>>> cur.fetchall()
[(u'from beginner to master', u'laoqi',
u'python'), (u'physics', u'first', u'c')]
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> cur.close()
>>> conn.close()
```

□□□
https://docs.Python.org/2/library/sqlite3.html

□□□

□□□□□□□□□□□□□□□□□□□qiwsir@126.com,□□□□□

[illegible]

[illegible] excel[illegible]“[illegible]”[illegible] word [illegible]

> VisiCalc [illegible]II[illegible]·[illegible]Dan Bricklin[illegible]·[illegible]Bob Frankston[illegible]1979 [illegible] 10 [illegible]
>
> [illegible] Lotus 1-2-3[illegible] Lotus Software[illegible] 1983 [illegible] DOS [illegible] 100 [illegible]
>
> [illegible] 1982 [illegible]──Multiplan[illegible] CP/M [illegible] MS-DOS [illegible]Multiplan [illegible] Lotus 1-2-3[illegible]
>
> 1985 [illegible] Excel[illegible] Mac [illegible] 1987 [illegible] 11 [illegible] Windows [illegible] Excel [illegible] Windows [illegible] windows [illegible] Lotus1-2-3 [illegible] Windows [illegible] 1988 [illegible]Excel [illegible] 1-2-3[illegible]
>
> [illegible]Excel [illegible] Office [illegible] Microsoft Office Excel[illegible]
>
> [illegible] Excel [illegible] VisiCalc [illegible]
>
> [illegible]Lotus 1-2-3 [illegible]

> [illegible] Linux [illegible]google [illegible]

[illegible]Python [illegible]

openpyl

openpyl [illegible] Microsoft Excel 2007/2010 [illegible] Excel 2010 xlsx/xlsm/xltx/xltm [illegible]

[illegible]

[illegible] pip install

```
$ sudo pip install openpyxl
```

[illegible]

```
Successfully installed openpyxl jdcal
Cleaning up...
```

workbook [illegible] sheet

[illegible]

```
>>> from openpyxl import Workbook
```

[illegible] `Workbook()` [illegible]

```
>>> wb = Workbook()
```

□□□ Excel □□□□□□□□□□□□□ Excel□□□□□□□□□□□□□□□□□□ (workbook)□□□□□□□□□ sheet□□□□□□□□□□□□□□□□□□□□□□□□ sheet□

```
>>> ws = wb.active
```

□□□□□□□□□□□□□ sheet□□□□□□□□□□□□□□□□□□□□□ sheet□□□□ □□□□□□□□□

□□□□□□ sheet □□□□□

```
>>> ws1 = wb.create_sheet()
```

□□□□□□□□□

```
>>> ws2 = wb.create_sheet(1)
```

□□□□□□□□□

□ Excel □□□□□□□□□ sheet □□□□□□□□"Sheet1"□"Sheet2"□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> ws.title = "Python"
```

ws□□□□sheet□□□□□□"Python"□□

□□□□□□□□□□□□□□□□□□□□□ sheet

```
>>> ws01 = wb['Python']    #sheet □□□□□□□□□□□□□□
□□
>>> ws is ws01
True
```

□□□□□□□

```
>>> ws02 = wb.get_sheet_by_name("Python")    #□□
□□□□□□□□□□□□□□□□□ sheet □□
```

```
>>> ws is ws02
True
```

[illegible](wb)[illegible] sheet[illegible]

```
>>> print wb.get_sheet_names()
['Python', 'Sheet2', 'Sheet1']
```

Sheet2 [illegible] sheet [illegible] Excel [illegible] sheet [illegible]"Sheet1"[illegible]

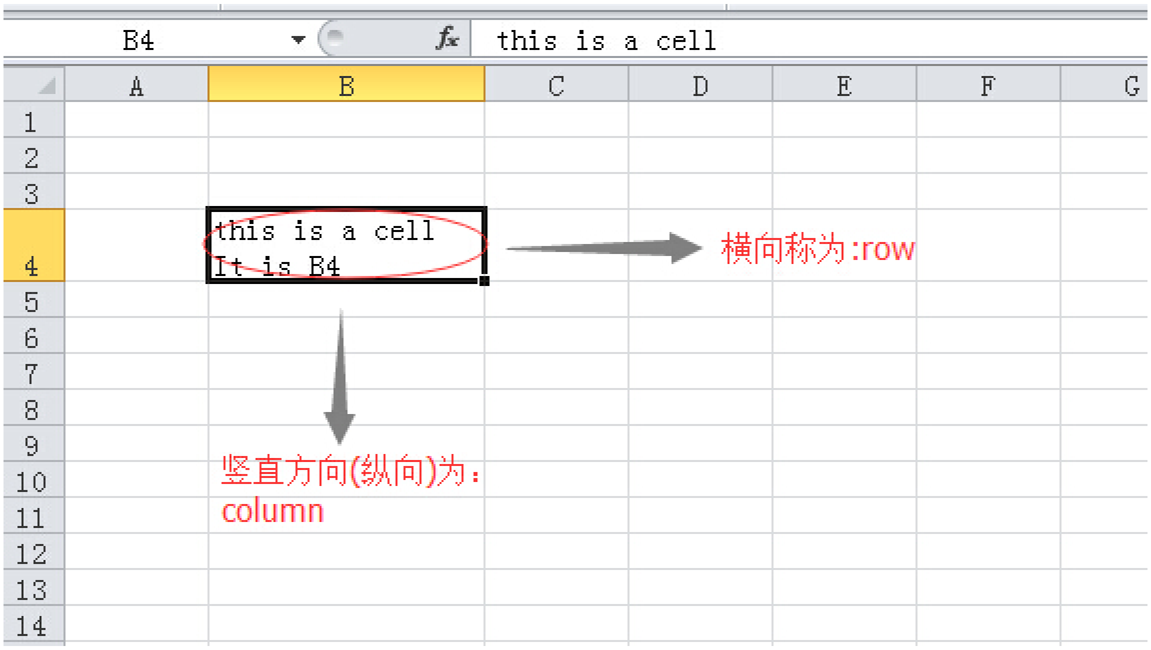
[illegible] sheet [illegible]

```
>>> for sh in wb:
...     print sh.title
...
Python
Sheet2
Sheet1
```

[illegible] `dir(wb)` [illegible]`__iter__`[illegible]

cell

[illegible]


B4
fx
this is a cell
A
B
C
D
E
F
G
1
2
3
4
5
6
7
8
9
10
11
12
13
14
this is a cell
It is B4
横向称为:row
竖直方向(纵向)为:
column

□□ sheet□□□□ cell □□□□□□□□□□□□□□□□ cell□□□□□□

```
b4 = ws['B4']
```

□□ B4 □□ cell □□□□□□□□□□□□□□□□□□□□□ b4□□□ sheet □□□□□ cell□□□□□□□□ cell □□□

□□□□□□□□□□□ Excel□□□□□□□□□□ cell□□□□□ Python □□□□□□□ □□ B4 □□ cell□□□□□□□□□□□□ b4 □□□□□

□□□□ B4 □□□□□□□□□□□

```
>>> ws['B4'] = 4444
```

□□ b4 □□□□□ cell □□□□□□□□□□□□□□□□□□□□□□

```
>>> b4.value
4444
```

□□□□□□□□□□□□□□ cell □□□□□□□□□□□□□

```
>>> a1 = ws.cell("A1")
```

□□□

```
>>> a2 = ws.cell(row = 2, column = 1)
```

□□□□□□□□□□□ sheet □□□□□□□□□□□ cell□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□

|A1|B1|C1| |A2|B2|C2| |A3|B3|C3|

□□□□□□□□□□□□□

```
>>> cells = ws["A1":"C3"]
```

□□□□□□□□□□□□□□

```
>>> tuple(ws.iter rows("A1:C3"))
((<Cell python.A1>, <Cell Python.B1>, <Cell
Python.C1>),
 (<Cell python.A2>, <Cell Python.B2>, <Cell
Python.C2>),
 (<Cell python.A3>, <Cell Python.B3>, <Cell
Python.C3>))
```

□□□□□□□□□□□□□□□ A1-B1-C1□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□ cell □□□

```
>>> for row in ws.iter rows("A1:C3"):
...     for cell in row:
...         print cell
...
<Cell Python.A1>
<Cell Python.B1>
<Cell Python.C1>
<Cell Python.A2>
<Cell Python.B2>
<Cell Python.C2>
<Cell Python.A3>
```

```
<Cell Python.B3>
<Cell Python.C3>
```

□□□□ sheet □□□ `rows` □□□□□□□□□□□□□□□□ cell □□□□□□□□□□□□□□□□□□□□□□□□□□ sheet □□□□ cell□□□□□□□□□□□□ B4□□□□□□□□□□□□□□ row□□

```
>>> ws.rows
((<Cell python.A1>, <Cell python.B1>, <Cell
python.C1>),
 (<Cell python.A2>, <Cell python.B2>, <Cell
python.C2>),
 (<Cell python.A3>, <Cell python.B3>, <Cell
python.C3>),
 (<Cell python.A4>, <Cell python.B4>, <Cell
python.C4>))
```

□ sheet □□□ `columns` □□□□□□□□□□□□□□□□ cell □□□□□□□□□□□

```
>>> ws.columns
((<Cell python.A1>, <Cell python.A2>, <Cell
python.A3>, <Cell python.A4>),
 (<Cell python.B1>, <Cell python.B2>, <Cell
python.B3>, <Cell python.B4>),
 (<Cell python.C1>, <Cell python.C2>, <Cell
python.C3>, <Cell python.C4>))
```

□□□□□□□□□□□□□ cell □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ 1,2,3,...

```
>>> i = 1
>>> for cell in ws.rows:
...     cell.value = i
...     i += 1
... Traceback (most recent call last): File "", line 2, in
AttributeError: 'tuple' object has no attribute 'value'
```

[illegible]tuple [illegible] tuple [illegible] cell [illegible]“[illegible]”[illegible]

```
>>> for row in ws.rows:
...     for cell in row:
...         cell.value = i
...         i += 1
...
```

[illegible] cell [illegible]

```
>>> for col in ws.columns:
...     for cell in col:
...         print cell.value
...
1
4
7
10
2
5
8
11
3
6
9
12
```

[illegible]

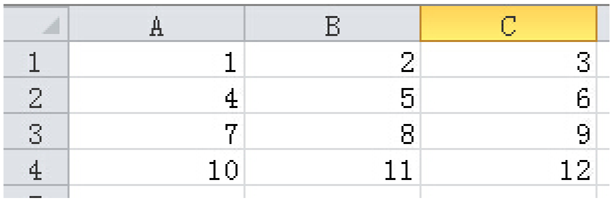
[illegible]

□□□□□□□□□□□□□□

```
>>> wb.save("23401.xlsx")
```

□□□□□□□□□□□□□□

□□□□□□ Excel □□□□□□□□□□□□□□□□



	A	B	C
1	1	2	3
2	4	5	6
3	7	8	9
4	10	11	12

□□□□□□

□□□□□□□ .xlsx □□□□□□□□□□□□□□□

```
>>> from openpyxl import load_workbook
>>> wb2 = load_workbook("23401.xlsx")
>>> print wb2.get_sheet_names()
['python', 'Sheet2', 'Sheet1']
>>> ws_wb2 = wb2["python"]
>>> for row in ws_wb2.rows:
...     for cell in row:
...         print cell.value
...
1
2
3
4
5
6
7
8
9
```

```
10
11
12
```

[illegible]

[illegible]

[illegible] openpyxl [illegible]

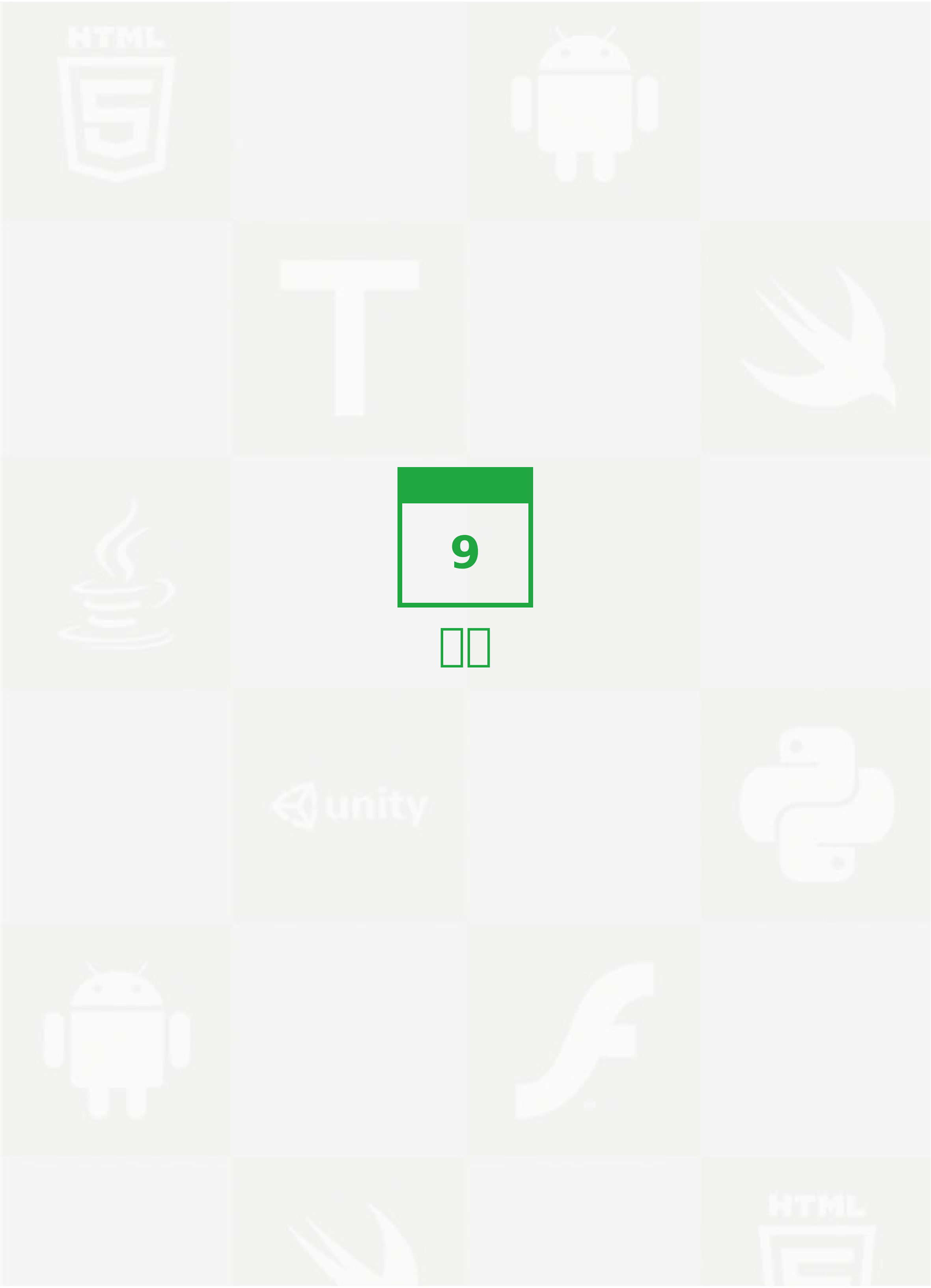
- xlsxwriter[illegible] Excel 2010 [illegible] .xlsx[illegible] https://xlsxwriter.readthedocs.org/[illegible]

[illegible] .xls [illegible]

- xlrd[illegible] https://secure.simplistix.co.uk/svn/xlrd/trunk/xlrd/doc/xlrd.html?p=4966
- xlwt[illegible]http://xlwt.readthedocs.org/en/latest/

[illegible]

[illegible]**qiwsir@126.com**,[illegible]



9

[illegible]

[illegible]

[illegible] Python [illegible]

[illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

10

□Tornado□□□

[illegible]

[illegible]

> [illegible]History of the World Wide Web

[illegible] ubuntu [illegible]

[illegible] ubuntu [illegible] Python [illegible]

[illegible] tornado [illegible]

[illegible]

[illegible]

[illegible]

> [illegible]Software framework[illegible]
>
> [illegible]

> [illegible]

[illegible]

[illegible]

Python [illegible]

[illegible] php([illegible] php[illegible]php [illegible]Python [illegible] web [illegible] web [illegible]

- Django:[illegible]
- Flask[illegible] Python [illegible] Web [illegible] Werkzeug WSGI [illegible] Jinja2 [illegible]
- Web2py[illegible] Python [illegible] Web [illegible] Web [illegible] Google App Engine[illegible]
- Bottle: [illegible] Python Web [illegible] WSGI[illegible] Python [illegible]
- Tornado[illegible] Tornado Web Server[illegible] Web [illegible] Python Web [illegible] FriendFeed [illegible]FaceBook [illegible]
- webpy: [illegible] Python Web [illegible]webpy [illegible]Keep it simple and powerful[illegible] URL [illegible]

□□□□□□□□□□http://blog.jobbole.com/72306/□□□□□□□□□□□□□
□□□□ web □□□□□□□□□□□□□□□□□

Tornado

□□□□□□□□□ Tornado □□□□□□□□□□□□ Django□□□□□□□□□□□□□□
□□□□□ Tornado,□□□□□□□......□□□□□□□□□□□□□□□

Tornado □□ Tornado Web Server□□□□□ Python □□□□□ Web □
□□□ Web □□□□□□ FriendFeed □□□□□□□□ FriendFeed □□□□□
Facebook □□□□□□□□□□□□□□□□□□□□ Tornado □□□□□□□□□□□□
□□□□□ Facebook□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□ Tornado □□□□□□□□□□□□□□□□□□□
□□

Tornado □□□□□□□□□□□□□□□□□□□□□□“C10k”□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□(□□ C10K □□□□□□□□□□C10k problem)

□□□□□□□□ Web □□□□□□□□□□□□□□□□□□□□□
https://developers.facebook.com/blog/post/301 □

□□□□□□□ AMD Opteron, □□ 2.4GHz, 4 □

□□	□□	□□/□□
Tornado	nginx, 4 □□	8213
Tornado	1 □□□□□□	3353
Django	Apache/mod_wsgi	2223
web.py	Apache/mod_wsgi	2066

[illegible]	[illegible]	[illegible]/[illegible]
CherryPy	[illegible]	785

[illegible] Tornado [illegible]

[illegible]——**Tornado**

[illegible] Tornado

Tornado [illegible]http://www.tornadoweb.org

[illegible]

```
pip install tornado
```

[illegible] Tornado [illegible] PyPI[illegible] pip [illegible] easy_install [illegible]

[illegible]
https://pypi.python.org/pypi/tornado/

[illegible] github [illegible] github [illegible]

[illegible] windows [illegible]

> Tornado will also run on Windows, although this configuration is not officially supported and is recommended only for development use.

[illegible] Linux [illegible]

[illegible]

□□□□□□□□□□□□□□□□□□□□

- HTML
- CSS
- JavaScript

□□□□□□□□□□□□□□□□□□□ JavaScript(JS) □□□□□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

□□ Hello

□□□□ Python □□ hello.py(□□□□□□□□)□

```
#!/usr/bin/env Python
#coding:utf-8

import tornado.httpserver
import tornado.ioloop
import tornado.options
import tornado.web

from tornado.options import define, options
define("port", default=8000, help="run on the
given port", type=int)

class IndexHandler(tornado.web.RequestHandler):
    def get(self):
        greeting = self.get_argument('greeting',
'Hello')
        self.write(greeting + ', welcome you to
read: www.itdiffer.com')

if   name   == "  main  ":
    tornado.options.parse command line()
    app = tornado.web.Application(handlers=
[(r"/", IndexHandler)])
    http server =
tornado.httpserver.HTTPServer(app)
```

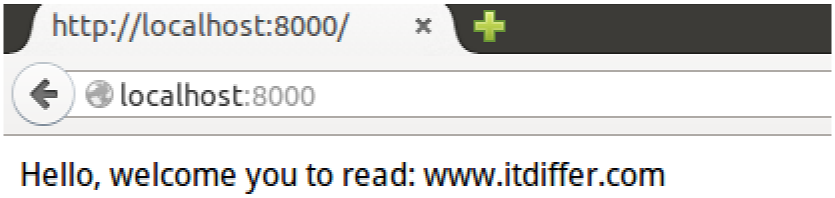
```
    http_server.listen(options.port)
    tornado.ioloop.IOLoop.instance().start()
```

□□□□□ hello.py □□□□□□□□□

```
$ python hello.py
```

□ Python □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□http://localhost:8000□□□□□□□□



Hello, welcome you to read: www.itdiffer.com

□□ ubuntu □ shell □□□□□□□□□□□□

```
 $ curl http://localhost:8000/
Hello, welcome you to read: www.itdiffer.com

$ curl http://localhost:8000/?greeting=Qiwsir
Qiwsir, welcome you to read: www.itdiffer.com
```

□□□□□□□□□□□□□□□□□

[illegible] Tornado [illegible] Tornado[illegible] Tornado [illegible] server[illegible]

[illegible] tornado [illegible]

WEB [illegible]

[illegible] Web [illegible]

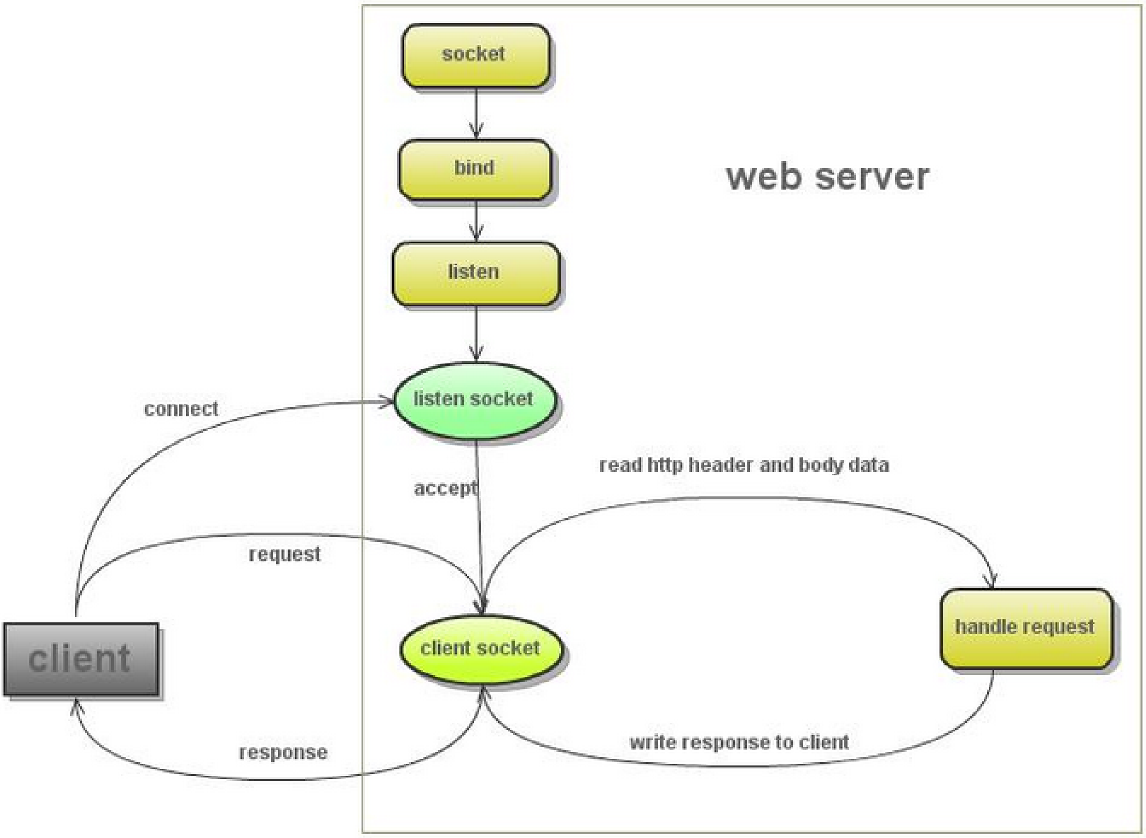
> [illegible] Web [illegible] Apache [illegible]

[illegible]

[illegible] Web [illegible] WEB [illegible]

[illegible] WEB [illegible] http [illegible] google.

[illegible]web[illegible]


socket
bind
web server
listen
connect
listen socket
accept
read http header and body data
request
client socket
handle request
client
response
write response to client

[illegible]

1. [illegible] listen socket, [illegible], [illegible]
2. listen socket [illegible], [illegible] client socket, [illegible] client socket [illegible]
3. [illegible], [illegible] client socket [illegible] http [illegible], [illegible] post [illegible], [illegible], [illegible], [illegible], [illegible] client socket [illegible]

[illegible]

```
import tornado.httpserver
import tornado.ioloop
import tornado.options
import tornado.web
```

[illegible] Tornado [illegible]

- tornado.httpserver[illegible] web [illegible] http [illegible]Tornado [illegible]
- tornado.ioloop[illegible] socket [illegible]
- tornado.options[illegible]
- tornado.web[illegible] Web [illegible]

[illegible] google [illegible]

[illegible] from...import [illegible]

```
from tornado.options import define, options
define("port", default=8000, help="run on the
given port", type=int)
```

[illegible]“[illegible]”[illegible] `tornado.options.define()` [illegible] `http:localhost:8000` [illegible] http [illegible] 80[illegible] 8000,[illegible] Nginx[illegible]Apache[illegible] 80,[illegible] 80[illegible] tornado [illegible] Nginx [illegible]

[illegible]-[illegible]

```
class IndexHandler(tornado.web.RequestHandler):
    def get(self):
        greeting = self.get_argument('greeting', 'Hello')
        self.write(greeting + ', welcome you to read: www.itdiffer.com')
```

□□“□□□□□”□□□

□□□□□□□□□□□□□□□□□□ IndexHandler□□□ Handler □□□□□□□□□□□□□□□□□□□□□□□□

□ IndexHandler □□ `tornado.web.RequestHandler`,□□□□□□ `get()` □ `post()` □□□ web □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□[HTTP GET POST □□□□□□□□□ href="https://github.com/qiwsir/ITArticles/blob/master/Tornado/DifferenceHttpGetPost.md")□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□ `get()` □□□□

□ `greeting = self.get_argument('greeting', 'Hello')` □□□□□□□□□ url □□□□□□□□□□□□

```
$ curl http://localhost:8000/?greeting=Qiwsir
Qiwsir, welcome you to read: www.itdiffer.com
```

□□□□□□ url □□□ greeting □□□□□ Qiwsir□□□□ url □□□□□□□□□□□ Hello.

□□□□□□□□□□□□□□□□□□□

> RequestHandler.get_argument(name, default=, []strip=True)
>
> Returns the value of the argument with the given name.

> If default is not provided, the argument is considered to be required, and we raise a MissingArgumentError if it is missing.
>
> If the argument appears in the url more than once, we return the last value.
>
> The returned value is always unicode.

[illegible] `self.write(greeting + ',weblcome you to read: www.itdiffer.com)'` [illegible] `write()` [illegible]

> RequestHandler.write(chunk)[source]
>
> Writes the given chunk to the output buffer.
>
> To write the output to the network, use the flush() method below.
>
> If the given chunk is a dictionary, we write it as JSON and set the Content-Type of the response to be application/json. (if you want to send JSON as a different Content-Type, call set_header after calling write()).

main() [illegible]

`if __name__ == "__main__"`,[illegible] Python [illegible]

`tornado.options.parse_command_line()`,[illegible] tornado [illegible] tornado [illegible] import [illegible] main[illegible]

Application □

□□□□□□□□

```
app = tornado.web.Application(handlers=[(r"/",
IndexHandler)])
```

□ tornado.web.Application □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ HTTPServer □□□□□□□□□ http □□□□□□□□ Application □□ `__init__` □□□□□□□

```
def   init  (self, handlers=None,
default host="", transforms=None,**settings):
    pass
```

□□□□□□□handlers □□□□□□□□□ Application □□□□□□□□□□□□□□□ □□□□□□□□□□ `handlers=[(r"/", IndexHandler)]` □□□□□□□ □□□□□□□□□□□□□□□ `http://localhost:8000` □□□□□□□□□ `http://localhost:8000/qiwsir` □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□ IndexHandler □□□□□□□□

□□ handlers □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□ list□list □□□□□□ tuple□tuple □□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□(□□□□□□□□□□□□□□□□□□)□□□□ □□

```
handlers = [
    (r"/", IndexHandlers),              #□□□□□□□
□□ IndesHandlers □□
    (r"/qiwsir/(.*)", QiwsirHandlers),  #□□
/qiwsir/ □□□□□□□□□□□□□□□□□□□□□□□□ QiwsirHandlers
□□
]
```

□□

[illegible] `r"/"` [illegible]r [illegible]\n[illegible]“\”[illegible](1)[illegible] `r"\n"` [illegible]\ [illegible] n[illegible] \ [illegible]'r'[illegible]

[illegible] Application [illegible]

HTTPServer [illegible]

[illegible]Application [illegible]app[illegible] HTTPServer [illegible]

```
http_server = tornado.httpserver.HTTPServer(app)
```

HTTPServer [illegible] tornado.httpserver [illegible]HTTPServer [illegible] HTTP [illegible] HTTPServer [illegible] Application [illegible],[illegible]options.port [illegible] IndexHandler [illegible] from...import.. [illegible]

```
http_server.listen(options.port)
```

[illegible] http [illegible]

[illegible]
http://tornado.readthedocs.org/en/latest/httpserver.html#http-server

IOLoop [illegible]

[illegible]

```
tornado.ioloop.IOLoop.instance().start()
```

[illegible]`__main()__`[illegible] HTTP [illegible]

[illegible] hello.py [illegible] Tornado [illegible]

[illegible]

[illegible]

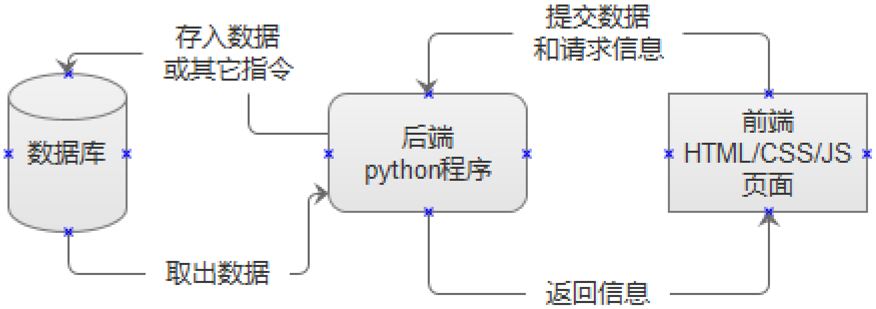
[illegible]**qiwsir@126.com**,[illegible]

[illegible] tornado [illegible] (1)

[illegible]

[illegible]

[illegible]


存入数据
或其它指令
数据库
取出数据
后端
python程序
提交数据
和请求信息
前端
HTML/CSS/JS
页面
返回信息

[illegible]

[illegible] HTML/CSS/JS[illegible] Python [illegible] HTML/CSS/JS[illegible]

MVC [illegible]

> MVC [illegible]Model-View-Controller[illegible]Model[illegible]View[illegible]Controller[illegible]

> MVC [illegible] Trygve Reenskaug [illegible] 1978 [illegible]Xerox PARC[illegible] 20 [illegible] 80 [illegible] Smalltalk [illegible]MVC [illegible]
>
> - [illegible] Controller[illegible]- [illegible]
> - [illegible] View[illegible] - [illegible] -[illegible] Model[illegible] - [illegible]([illegible])[illegible]

[illegible]“[illegible]”[illegible] View [illegible] MVC [illegible]

[illegible]

- [illegible] url[illegible]
- [illegible]

[illegible]

[illegible] MVC [illegible] Controller [illegible] Model [illegible]“[illegible]”[illegible]

[illegible]

[illegible] Python [illegible]

[illegible]

[illegible] Python [illegible]

[illegible] Python [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
 /.
 |
handlers
 |
methods
 |
statics
 |
templates
 |
application.py
 |
server.py
 |
url.py
```

[illegible]

[illegible]

- handlers[illegible] Python [illegible]
- methods[illegible] handlers [illegible]

- statics□□□□□□□□□□□□□□□□□□css □ javascript □□□□
- templates□□□□□□□□□□□□ html □□□□□□□□□□□□□□□□

□□□□□□□ Python □□□

url.py □□

```
#!/usr/bin/env Python
# coding=utf-8
"""
the url structure of website
"""

import sys     #utf-8□□□□□
reload(sys)
sys.setdefaultencoding("utf-8")

from handlers.index import IndexHandler    #□□□□
□□

url = [
    (r'/', IndexHandler),
]
```

url.py □□□□□□□□□□□□□□□`from handlers.index import IndexHandler`□□□□ handlers □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ handlers □□□□□□□□□□□□□□ index.py□□□□□□□□□ IndexHandler□□ url.py □□□□□□□□□□□□

□□ url □□□□□□□□□□□□□□□□□□□□□□□□□□□□ `(r'/', IndexHandler),`□□□□□□□□□□□□□□□□ IndexHandler□□□□□□□□□ get() □□ post() □□□□□ IndexHandler □□□□□□□□□□

□□□□□□□□□□□□□□

application.py □□

```
#!/usr/bin/env Python
# coding=utf-8

from url import url

import tornado.web
import os

settings = dict(
    template_path =
os.path.join(os.path.dirname(__file__),
"templates"),
    static_path =
os.path.join(os.path.dirname(__file__),
"statics")
    )

application = tornado.web.Application(
    handlers = url,
    **settings
    )
```

□□

`from url import url` □□ url.py □□□□□□□□□□□

setting □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□"templates"□"statics"□□□□□□□□□□□□□□□

□□□□ application □□□□□□□□□□□□□□□□
`tornado.web.Application()` □□□□□□

> tornado.web.Application(handlers=None, default_host='', transforms=None, **settings)

[illegible] settings [illegible] `debug = True` [illegible] settings [illegible][tornado.web-RequestHandler and Application classes](#)

server.py [illegible]

[illegible] tornado [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.ioloop
import tornado.options
import tornado.httpserver

from application import application

from tornado.options import define, options

define("port", default = 8000, help = "run on
the given port", type = int)

def main():
    tornado.options.parse_command_line()
    http server =
tornado.httpserver.HTTPServer(application)
    http_server.listen(options.port)

    print "Development server is running at
http://127.0.0.1:%s" % options.port
    print "Quit the server with Control-C"

    tornado.ioloop.IOLoop.instance().start()
```

```
if __name__ == "__main__":
    main()
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible] tornado [illegible] (2)

[illegible]

[illegible]

[illegible]

[illegible] mysql [illegible]

[illegible] mysql [illegible][illegible]mysql [illegible](1)[illegible][illegible]

[illegible] methods [illegible] db.py[illegible][illegible]mysql [illegible](1)[illegible][illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import MySQLdb

conn = MySQLdb.connect(host="localhost",
user="root", passwd="123123", db="qiwsirtest",
port=3306, charset="utf8")    #[illegible]

cur = conn.cursor()    #[illegible]
```

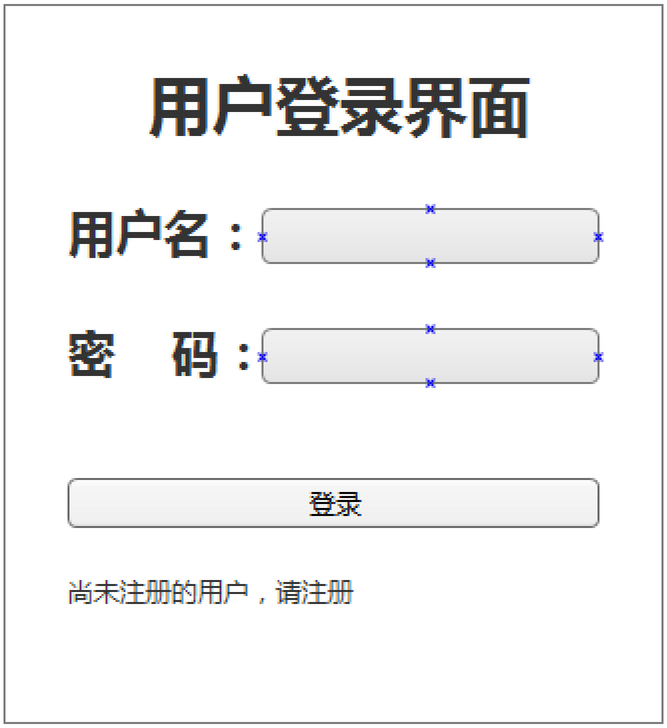
[illegible]

[illegible]

[illegible]

> [illegible]

[illegible]



用户登录界面

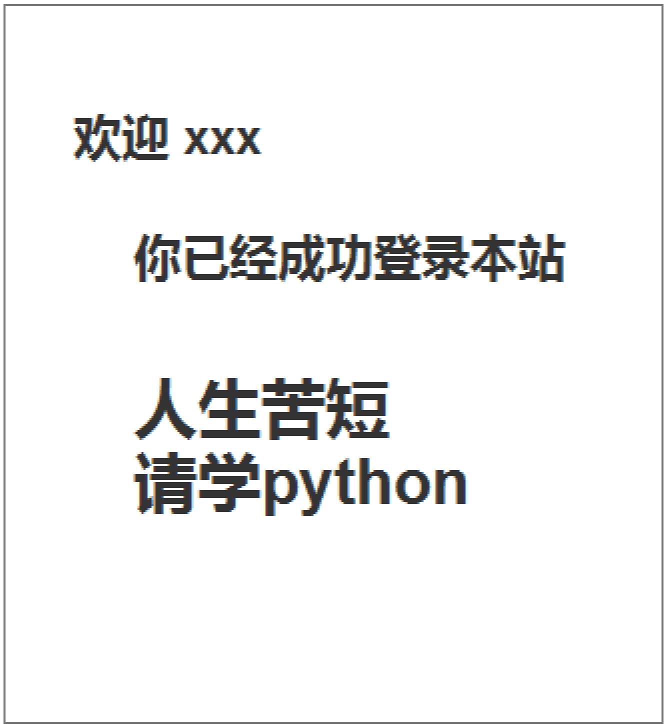
用户名：

密　码：

登录

尚未注册的用户，请注册

[illegible]“[illegible]”[illegible]


欢迎 xxx

你已经成功登录本站

人生苦短
请学python

□□ HTML □□□□□□□□□□□ templates □□□□□□□ index.html □□□□

```
<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <title>Learning Python</title>
</head>
<body>
    <h2>Login</h2>
    <form method="POST">
        <p><span>UserName:</span><input
type="text" id="username"/></p>
        <p><span>Password:</span><input
type="password" id="password" /></p>
        <p><input type="BUTTON" value="LOGIN"
id="login" /></p>
```

```
    </form>
</body>
```

[illegible] `<meta name="viewport" content="width=device-width, initial-scale=1" />` [illegible](viewport)[illegible](width=device-width)[illegible] 1.0(initial-scale=1)[illegible] 100%[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible] CSS[illegible]JS [illegible]

> [illegible] HTML5 [illegible]
>
> HTML5 [illegible] HTML [illegible]2014 [illegible] 10 [illegible]W3C[illegible] 1999 [illegible] HTML 4.01 [illegible] XHTML 1.0 [illegible] HTML5 [illegible] HTML[illegible]CSS [illegible] JavaScript [illegible]
>
> [illegible]Responsive web design[illegible] RWD[illegible]

[illegible] .html [illegible]

[illegible] jQuery

[illegible] login [illegible]——javascript[illegible]

> [illegible] javascript[illegible] java [illegible] javascript [illegible] java [illegible]“[illegible]”[illegible]“[illegible]”[illegible]

> JavaScript[illegible] JavaScript [illegible] HTML [illegible] HTML [illegible] JavaScript [illegible] Node.js[illegible]
>
> [illegible] 1995 [illegible]·[illegible] Java[illegible] JavaScript[illegible] Self [illegible] Scheme [illegible]
>
> [illegible] JScript[illegible] JavaScript [illegible]1997 [illegible] ECMA[illegible] Borland [illegible]ECMA-262[illegible] JavaScript [illegible] ECMA [illegible] ECMAScript[illegible]

[illegible] jQuery[illegible]

> jQuery [illegible] JavaScript [illegible] HTML [illegible] JavaScript [illegible]·[illegible]John Resig[illegible] 2006 [illegible] 1 [illegible] BarCamp NYC [illegible] Dave Methvin [illegible] 10,000 [illegible] 65% [illegible] jQuery[illegible] JavaScript [illegible]

[illegible] index.html [illegible] jQuery [illegible]

[illegible] HTML [illegible] jQuery [illegible] html [illegible] `<head>...</head>` [illegible] `</body>` [illegible]

[illegible]

```
<script
src="http://ajax.googleapis.com/ajax/libs/jquery
/1.11.2/jquery.min.js"></script>
```

[illegible] jQuery CDN(Content Delivery Network)[illegible]

[illegible]jQuery CDN [illegible]

```
<script src="//code.jquery.com/jquery-1.11.3.min.js"></script>
```

[illegible] jQuery [illegible]https://jqueryui.com/[illegible] statics [illegible] js[illegible]jquery [illegible] js [illegible] min.js [illegible]

[illegible] `statics/js` [illegible] jquery.min.js[illegible]

[illegible]

```
<script src="statics/js/jquery.min.js"></script>
```

[illegible] tornado [illegible]

```
<script src="{{static_url("js/jquery.min.js")}}"></script>
```

[illegible] jquery[illegible] js [illegible] script.js[illegible]

```
<script src="{{static_url("js/script.js")}}"></script>
```

[illegible] static_url [illegible] tornado [illegible] statics [illegible] statics [illegible]

[illegible] js

[illegible]

□□□□□□ statics/js/script.js □□□□□□□□□□□□□□□□□□□

```
$(document).ready(function(){
    alert("good");
    $("#login").click(function(){
        var user = $("#username").val();
        var pwd = $("#password").val();
        alert("username: "+user);
    });
});
```

□□□□□□□□□□□ javascript □□ jquery□□□□□ js □□□□□□□□□□□□□ □□□□□

□□□□□□□□□□□□□□ id □□□□ username □ password □□□□□□ alert □□□□□□□□□□□□□□□□□□□□□

hanlers □□□□□

□□□□□□□□□□□□ url.py □□□□□□□□□□□□

```
from handlers.index import IndexHandler    #□□□□
□□

url = [
    (r'/', IndexHandler),
]
```

□□□□□□□□□□□□□□□□□□□□□ handlers □□□□ index.py □□□□□□□ □□□□

```
#!/usr/bin/env Python
# coding=utf-8
```

```
import tornado.web

class IndexHandler(tornado.web.RequestHandler):
    def get(self):
        self.render("index.html")
```

[illegible] `localhost:8000` [illegible] `http://127.0.0.1:8000` [illegible] handlers [illegible] index.py [illegible] IndexHandler [illegible] get() [illegible] index.html [illegible]

`render()` [illegible]

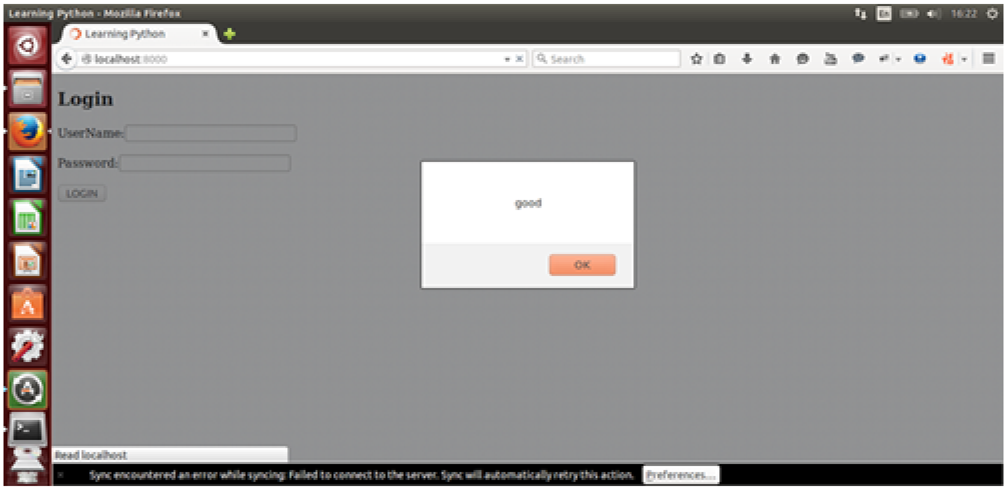
[illegible] handlers [illegible] `__init__.py` [illegible] Python [illegible] `__init__.py` [illegible] .py [illegible] Python [illegible]

[illegible] tornado [illegible] server.py [illegible]

```
$ python server.py
Development server is running at
http://127.0.0.1:8000
Quit the server with Control-C
```

[illegible] shell [illegible]

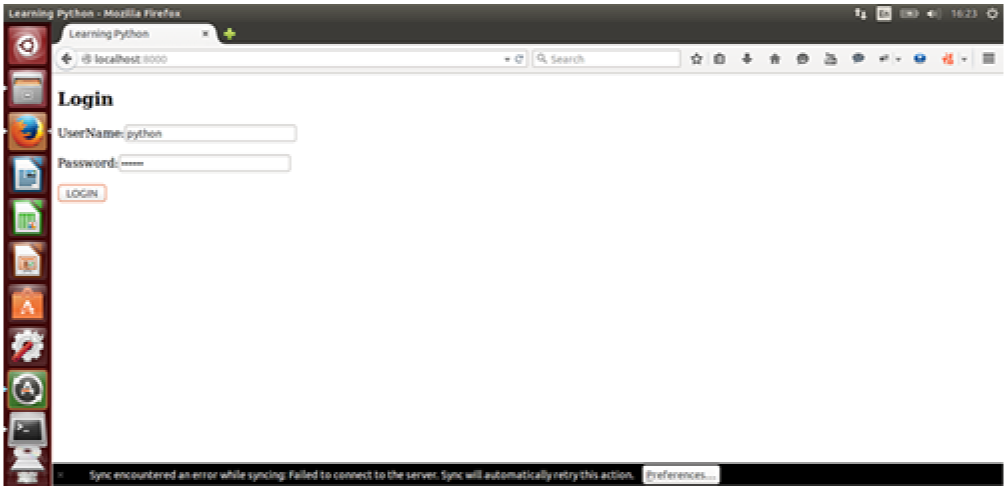
[illegible] `http://localhost:8000` [illegible] `http://127.0.0.1:8000` [illegible]


Learning Python - Mozilla Firefox
Learning Python
localhost:8000
Search
Login
UserName:
Password:
LOGIN
OK
Read localhost
Preferences...

good

Sync encountered an error while syncing: Failed to connect to the server. Sync will automatically retry this action.

□□□ script.js □□□□□□□□□□□□□□□□□□□□□□□□“□□”□□□□□□□□

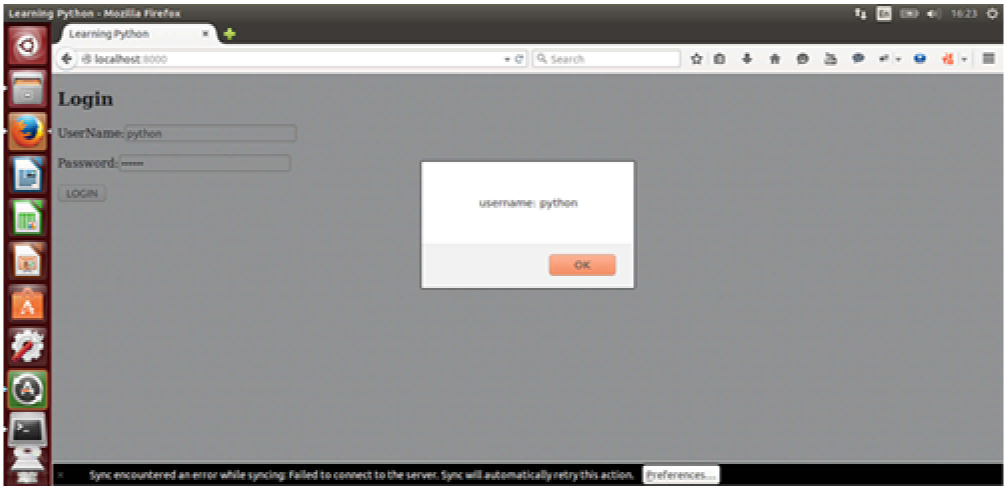

Learning Python - Mozilla Firefox
Learning Python
localhost:8000
Search
Login
LOGIN
Preferences...

UserName: python

Password: ••••••

Sync encountered an error while syncing: Failed to connect to the server. Sync will automatically retry this action.

□□□□□□□□□□□□□□□□□□ Login □□□□□□


Learning Python - Mozilla Firefox
Learning Python
localhost:8000
Search
Login
LOGIN
OK
Preferences...

UserName: python

Password: ••••••

username: python

Sync encountered an error while syncing: Failed to connect to the server. Sync will automatically retry this action.

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□ tornado □□□ (3)

□□□□

□□ ajax() □□□□

□□ ajax □□□□□□□□□□□□□□□□□□□

ajax □“Asynchronous Javascript and XML”□□□ JavaScript □ XML□□□□□□□□□□□□□□□□□□□□□□□□□□□ IE □□□□□ XHR(XML HttpRequest) □□ web □□□□ web □□□□□□□□□□□□ W3C □□□□□□□□□□□□□□□ Ajax □□□□□□□□□□□□□□□□□□□□ Oddpost□□□□ Yahoo!□□□□□ Yahoo! Mail □□□□□□□□□□ Ajax □□□□□ google □□□□□□□□□ google □ Gmail□Suggest □ Maps □□□□□□□ Ajax□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□ jQuery□□□□□ ajax() □□□□□□□□□□□□□□□

> ajax() □□□□ HTTP □□□□□□□□□
>
> □□□□ jQuery □□ AJAX □□□□□□□□□□□□□ $.get, $.post □□ $.ajax() □□□□□□ XMLHttpRequest □□
>
> □□□□□□□□$.ajax() □□□□□□□□□□□□□

□□□□□ Ajax □□□□□□□□□□□□□□——“□□”□□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□□□□

> "[illegible]"[illegible]"[illegible]"[illegible]callback[illegible]
>
> "[illegible]"[illegible] Ajax [illegible]"[illegible]"[illegible] http [illegible]

[illegible]ajax() [illegible]

[illegible] ajax() [illegible] script.js [illegible]

```
$(document).ready(function(){
    $("#login").click(function(){
        var user = $("#username").val();
        var pwd = $("#password").val();
        var pd = {"username":user,
"password":pwd};
        $.ajax({
            type:"post",
            url:"/",
            data:pd,
            cache:false,
            success:function(data){
                alert(data);
            },
            error:function(){
                alert("error!");
            },
        });
    });
});
```

[illegible]`var pd = {"username":user, "password":pwd};`[illegible] user [illegible] pwd [illegible] json [illegible]

json□□□□□□□□(8)□□□□□□□□ json □□□□□□□□□□ ajax() □□□□□
json □□□□□□□

jQuery □□ ajax() □□□□□□□□□□□□□□□□□□□□□ `$.ajax()` □□□□
□□□□□□□□□□□□□□

- type□post □□ get□□□ post □ get □□□□□□□□□[HTTP POST GET □□□□□□ href="https://github.com/qiwsir/ITArticles/blob/master/Tornado/DifferenceHttpGetPost.md")
- url□post □□ get □□□
- data□□□□□□□□□□□□□1□html □□□□□□□□2□json □□□□□3□form □□□ serialize() □□□□□□□□□□□□□□□□□ json □□□□□□□□□□□□□□□□
- cache□□□□ true□□□□□□□□□□□□ false.
- success□□□□□□□□□□□□□□□□□□□□□ data □ alert □□□□□□□□□□□□□□□□□□□□□□□□□□□□ alert() □□□
- error□□□□□□□□□□□□□□

□□□□□□

□□□□ ajax □□□□□□□ json □□□□□□□□□□□□□□□□□`"/"`□□□□□□ url.py □□□□□□□□□□□ handlers □□□ index.py □□□ IndexHandler □□□□□□□□□ post □□□□□□□□□□□□□□□□□ post □□□□□□□□□□□□□ IndexHandler □□□□ post()□□□□□□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.web

```

```
class IndexHandler(tornado.web.RequestHandler):
    def get(self):
        self.render("index.html")

    def post(self):
        username = self.get_argument("username")
        password = self.get_argument("password")
        self.write(username)
```

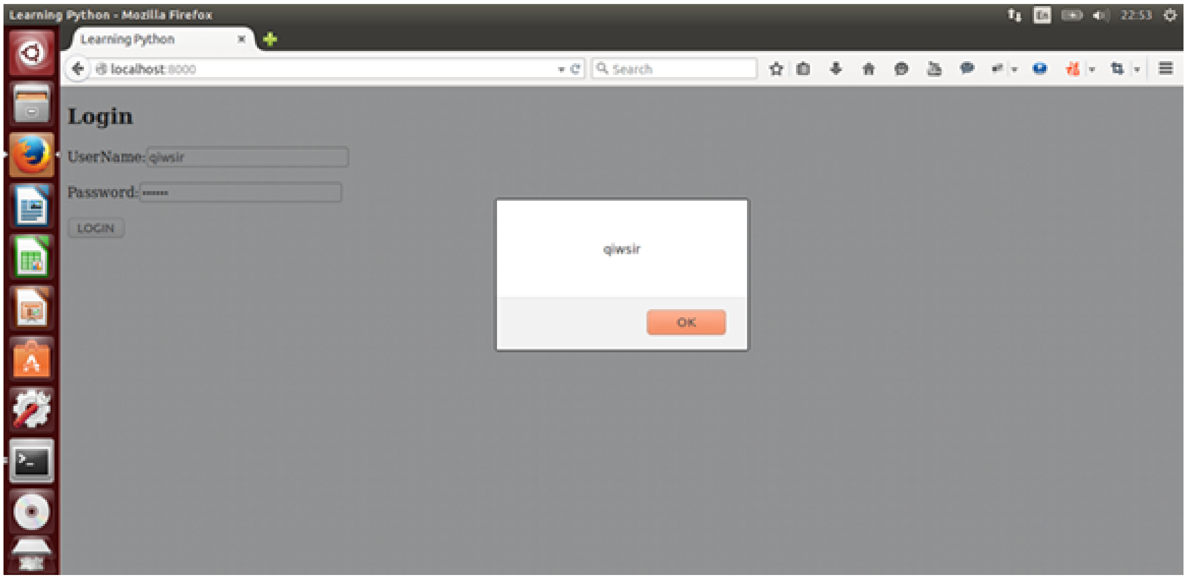
□ post() □□□□□□ get_argument() □□□□□□□□□□□□□□□□□□□□□□□□ `get_argument(name, default=[], strip=True)` □□□□□□ name □□□□□□□□□□□name □□□□□□□□□□□□ json □□□□□□□□□□□□□□□□□□□□□□□□□□ name □□□□□□ default □□□□□□□□□□□□□□□□□□□□□□□□□□□□ HTTP 400□□□□□□□□□ get □□□□□□ get_argument() □□□□ url □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `get_arguments(name, strip=true)` □

□□□□□□ get_argument() □□□□□ username □ password□□□□□□□ unicode □□□□□□

tornado.web.RequestHandler □□□ write()□□□□□□ `self.write(username)` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ json □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ctrl+c□□□□□□□ `Python server.py` □□□ setting □□□□□□ `debug = True` □□□□□□□□□□□□□□□ tornado □□□ (1)□□□

□□□□□□□□□□



Learning Python - Mozilla Firefox

localhost:8000

Login

UserName: qiwsir

Password: •••••••

□□□□□□□□□□□□□□□□□□ login □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□ ajax □□□□□□□□□□□□ get_argument() □□□

- □□□□□□□□□□□
- □□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□ methods □□□□□□□□□□ db.py□□□□□□□□□□□□ readdb.py□□□ db.py □□□□□□□□□□□□□

```
 #!/usr/bin/env Python
# coding=utf-8
```

```
from db import *

def select_table(table, column, condition, value
):
    sql = "select " + column + " from " + table
+ " where " + condition + "='" + value + "'"
    cur.execute(sql)
    lines = cur.fetchall()
    return lines
```

[illegible]

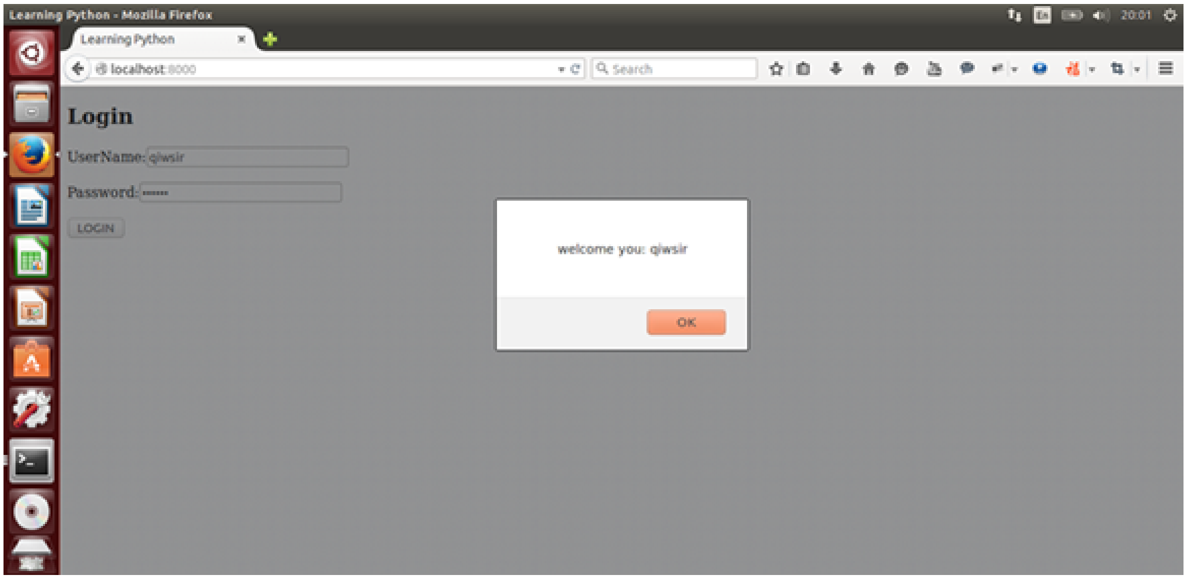
[illegible] index.py [illegible] post() [illegible] index.py [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.web
import methods.readdb as mrd

class IndexHandler(tornado.web.RequestHandler):
    def get(self):
        self.render("index.html")

    def post(self):
        username = self.get_argument("username")
        password = self.get_argument("password")
        user_infos =
mrd.select_table(table="users",column="*",condit
ion="username",value=username)
        if user_infos:
            db_pwd = user_infos[0][2]
            if db_pwd == password:
                self.write("welcome you: " +
username)
```

```
            else:
                self.write("your password was not right.")
        else:
            self.write("There is no thi user.")
```

□□□□□□ methods □□□□□□□□□ `__init__.py` □□□□□□ index.py □□□ `import methods.readdb as mrd`□

□□□□□□□□□□□□□□□



Learning Python - Mozilla Firefox

localhost:8000

Login

UserName: qiwsir

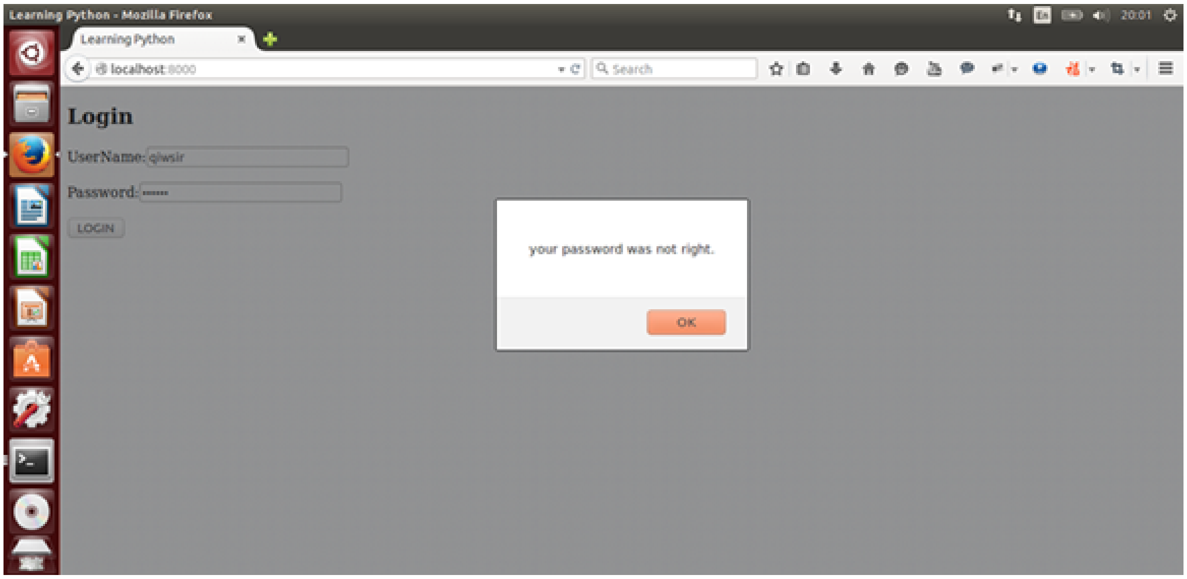
Password: ••••••

LOGIN

welcome you: qiwsir

OK

□□□


Learning Python - Mozilla Firefox
Learning Python
localhost:8000
Search
LOGIN
OK

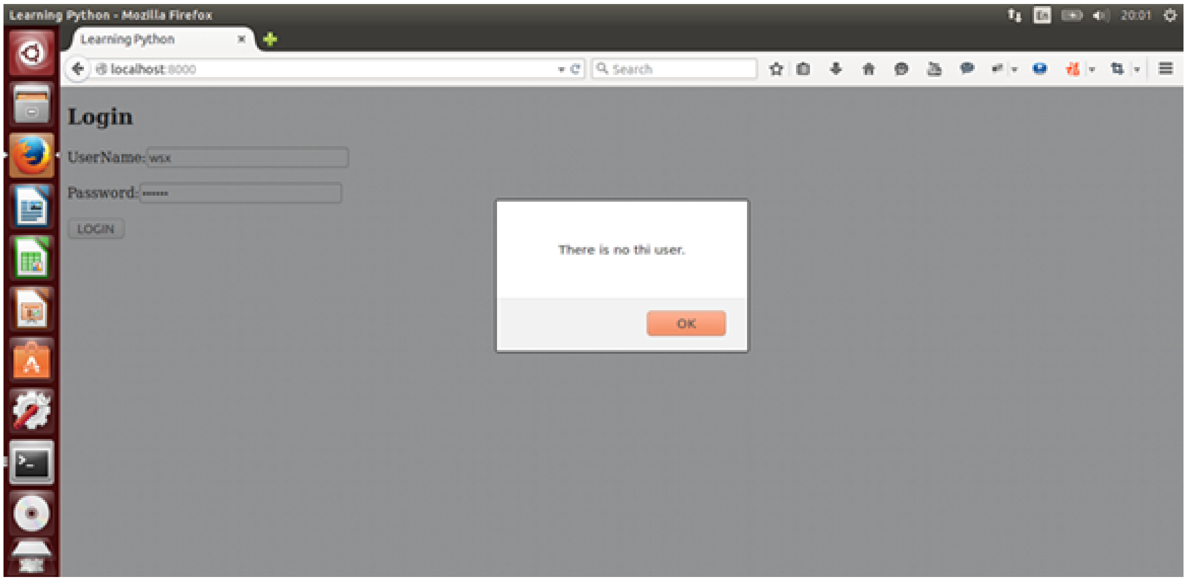
Login

UserName: qiwsir

Password: ••••••

your password was not right.

[illegible]


Learning Python - Mozilla Firefox
Learning Python
localhost:8000
Search
LOGIN
OK

Login

UserName: wsx

Password: ••••••

There is no thi user.

[illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible] tornado [illegible] (4)

[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

[illegible]tornado [illegible](tornado.template)[illegible]

render()

render() [illegible] tornado [illegible] IndexHandler [illegible] get() [illegible] `self.render("index.html")` [illegible] tornado [illegible] templates [illegible] index.html [illegible] index.html [illegible] index.html [illegible] html [illegible]“[illegible]”[illegible] index.html [illegible] index.py [illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.web
import methods.readdb as mrd

class IndexHandler(tornado.web.RequestHandler):
    def get(self):
        usernames =
mrd.select_columns(table="users",column="usernam
e")
        one_user = usernames[0][0]
        self.render("index.html", user=one_user)
```

index.py □□□□□□□□ get() □□□□□□□□□□□□□□□□□□□□□□
□one user□□□□□□ `self.render("index.html", user=one_user)` □□□□□□□□ index.html □□□□ `user=one_user` □□□□□□□□□□□□□

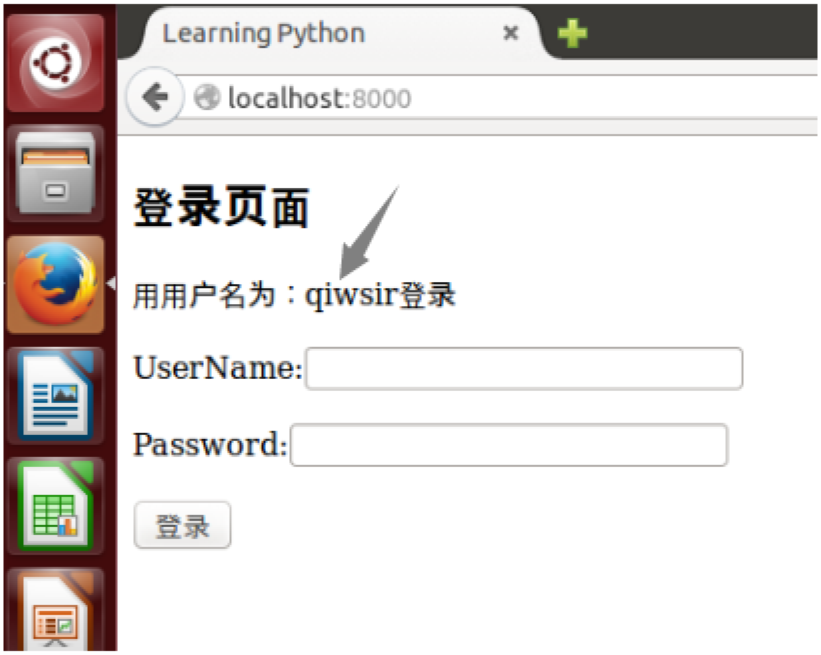
□□□□□□□□□□□□□□□□□□□□□
`mrd.select_columns(table="users",column="username")` □□□□□□□□□□ methods □□□□ readdb.py □□□□□□□□ select_columns □□□□□□□□□□□□□□□□□□□□□□□ readdb.py □□□ □□□□□□□□□□ select_columns□

```
#!/usr/bin/env Python
# coding=utf-8

from db import *

def select_table(table, column, condition, value
):
    sql = "select " + column + " from " + table
+ " where " + condition + "='" + value + "'"
    cur.execute(sql)
    lines = cur.fetchall()
    return lines

def select_columns(table, column ):
    sql = "select " + column + " from " + table
    cur.execute(sql)
    lines = cur.fetchall()
    return lines
```

□□□ index.html □□□□□□□

```
<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <title>Learning Python</title>
</head>
<body>
    <h2> □□□□</h2>
    <p>□□□□□□{{user}}□□</p>
    <form method="POST">
        <p><span>UserName:</span><input
type="text" id="username"/></p>
        <p><span>Password:</span><input
type="password" id="password" /></p>
        <p><input type="BUTTON" value="□□"
id="login" /></p>
    </form>
    <script src="
{{static_url("js/jquery.min.js")}}"></script>
    <script src="
{{static_url("js/script.js")}}"></script>
</body>
```

`<p> □□□□□□{{user}}□□</p>`□□□□□`{{ }}`□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□


Learning Python
localhost:8000

登录页面

用用户名为：qiwsir登录

UserName:

Password:

登录

[illegible] `{{ }}` [illegible]

`{{ }}` [illegible] html [illegible] qiwsir[illegible] render() [illegible] render() [illegible]

[illegible]

[illegible] url.py [illegible]

```
#!/usr/bin/env Python
# coding=utf-8
"""
the url structure of website
"""
import sys
reload(sys)
sys.setdefaultencoding("utf-8")

from handlers.index import IndexHandler
from handlers.user import UserHandler

url = [
    (r'/', IndexHandler),
    (r'/user', UserHandler),
]
```

□□□□□ handlers/user.py □□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.web
import methods.readdb as mrd

class UserHandler(tornado.web.RequestHandler):
    def get(self):
        username = self.get_argument("user")
        user_infos = 
mrd.select_table(table="users",column="*",condit
ion="username",value=username)
        self.render("user.html", users = 
user_infos)
```

□ get() □□□ `self.get_argument("user")` □□□□□□□ url □□□□ user □□□□□□□□□□□□□□□□□□□□□□□ js □□□□□□□□□□□□□□□

□□□□□□ user.py □□□□□□□□□□□□□□□□□□ user_infos □□□□□□□□□□□□□□□□□□□□□□□□□□ try...except□

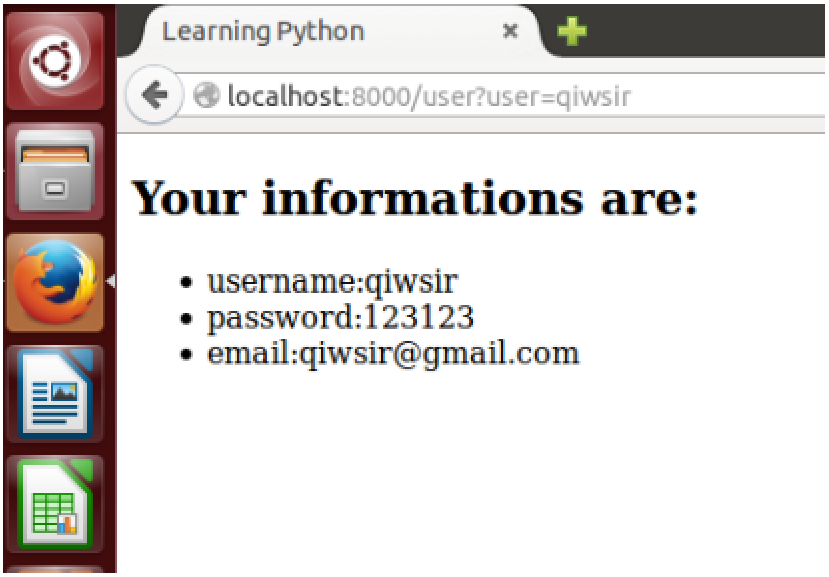
```
$(document).ready(function(){
    $("#login").click(function(){
        var user = $("#username").val();
        var pwd = $("#password").val();
        var pd = {"username":user, 
"password":pwd};
```

```
        $.ajax({
            type:"post",
            url:"/",
            data:pd,
            cache:false,
            success:function(data){
                window.location.href = "/user?
user="+data;
            },
            error:function(){
                alert("error!");
            },
        });
    });
});
```

□□□□ user.html □□□□□□□□□□□□user_infos □□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <title>Learning Python</title>
</head>
<body>
    <h2>Your informations are:</h2>
    <ul>
        {% for one in users %}
            <li>username:{{one[1]}}</li>
            <li>password:{{one[2]}}</li>
            <li>email:{{one[3]}}</li>
        {% end %}
    </ul>
</body>
```

□□□□□□□


Learning Python
localhost:8000/user?user=qiwsir

Your informations are:

- username:qiwsir
- password:123123
- email:qiwsir@gmail.com

□□□□□□□□□□□□□□□□□□

□□□□

□□□□□□□□□□□□□□□ Python □□□□□□□□□□□□

```
>>> from tornado.template import Template
>>> print Template("{{ 3+4 }}").generate()
7
>>> print Template("{{ 'python'[0:2] }}").generate()
py
>>> print Template("{{ '-'.join(str(i) for i in range(10)) }}").generate()
0-1-2-3-4-5-6-7-8-9
```

□□□□□□□□□□□□□□□`{{ 3+4 }}`□□□□□□□ render() □□□□□□□□□□□□□□□□□□□ `7` □□□□ tornado □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□Python □□□□□□□□□□□□□□□□ if□for□while □ try□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□ HTML □□□□

□□□□□□`{{% □□ %}}`

□□□

```
{{% if user=='qiwsir' %}}
    {{ user }}
{{% end %}}
```

□□□□□□□□□□□□□ if □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□ render □□□□□□□□□tornado □□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□ □□□Escape Sequence□□□□□□□(Character Entity)□□□□□□□□□ □□□ `<, >` □□□□□□□□□□□□□□□□□□□□□□□□□□ HTML □□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□ ASCII □□□□□□□□□□□□□“©”□□□ □□□□□□ HTML □□□□□□□□□□□□□□“©”□□□□□□□□“□copy;”□□

□□□

□□□□□□□□□□□□`select username, password from usertable where username='qiwsir'`□□□□□□□□□□□□ qiwsir□□□□□□ `a;drop database;` □□□□□□□□□□□ `select username, password from usertable where username=a; drop database;`□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□ `drop database`□□□□□□□□□□□□□□□□□□□□□□□□□□□ □ `<input type="text" />`□□□□□□□□□□□□□□□ `<form action="..."`□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ sql □□□□□□□□□□□

□□□□□□□□□□

□□□□□□□□□□

Tornado □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ Tornado □□□

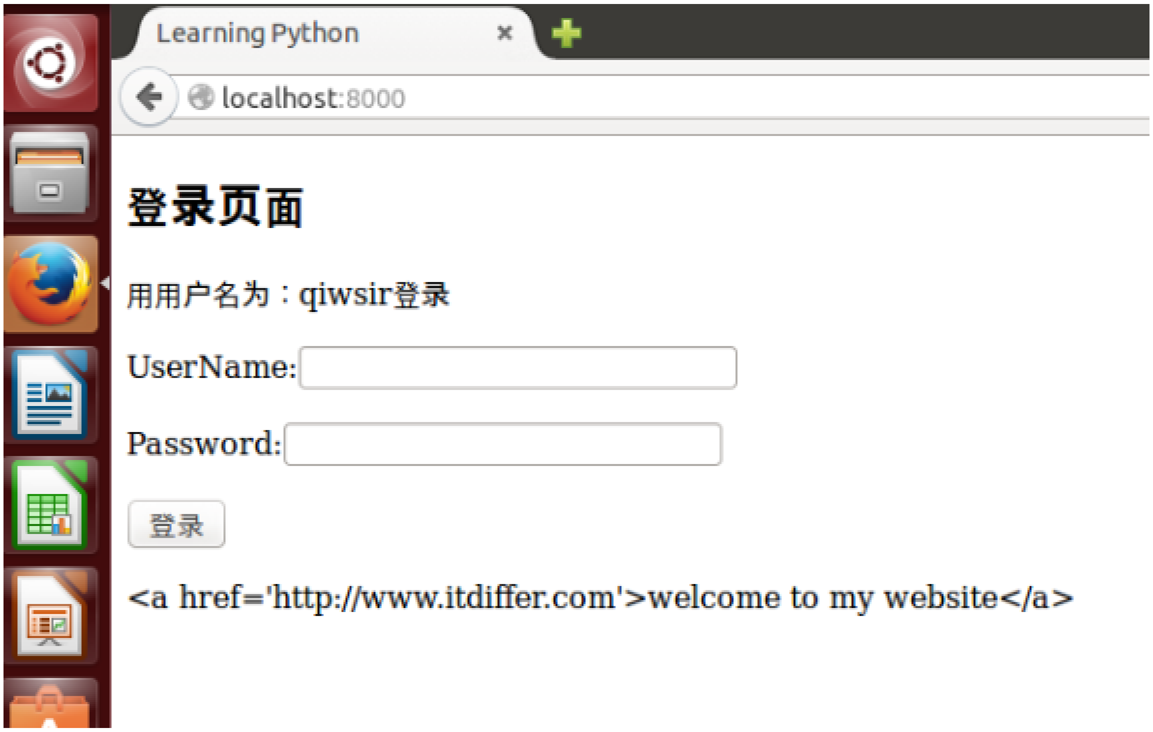
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ print □□□□□□□□□□□□□□

> print □□□□ Python3 □□ print() □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <title>Learning Python</title>
</head>
<body>
    <h2>□□□□</h2>
    <p>□□□□□□{{user}}□□</p>
    <form method="POST">
        <p><span>UserName:</span><input
type="text" id="username"/></p>
        <p><span>Password:</span><input
type="password" id="password" /></p>
        <p><input type="BUTTON" value="□□"
id="login" /></p>
    </form>
    {% set website = "<a
href='http://www.itdiffer.com'>welcome to my
website</a>" %}
    {{ website }}
    <script src="
{{static_url("js/jquery.min.js")}}"></script>
    <script src="
{{static_url("js/script.js")}}"></script>
</body>
```

[illegible] index.html [illegible] `{% set website = "<a href='http://www.itdiffer.com'>welcome to my website</a>" %}` [illegible] website[illegible] `{{ website }}` [illegible]“welcome to my website”[illegible]



Learning Python
localhost:8000

登录页面

用用户名为：qiwsir登录

UserName:

Password:

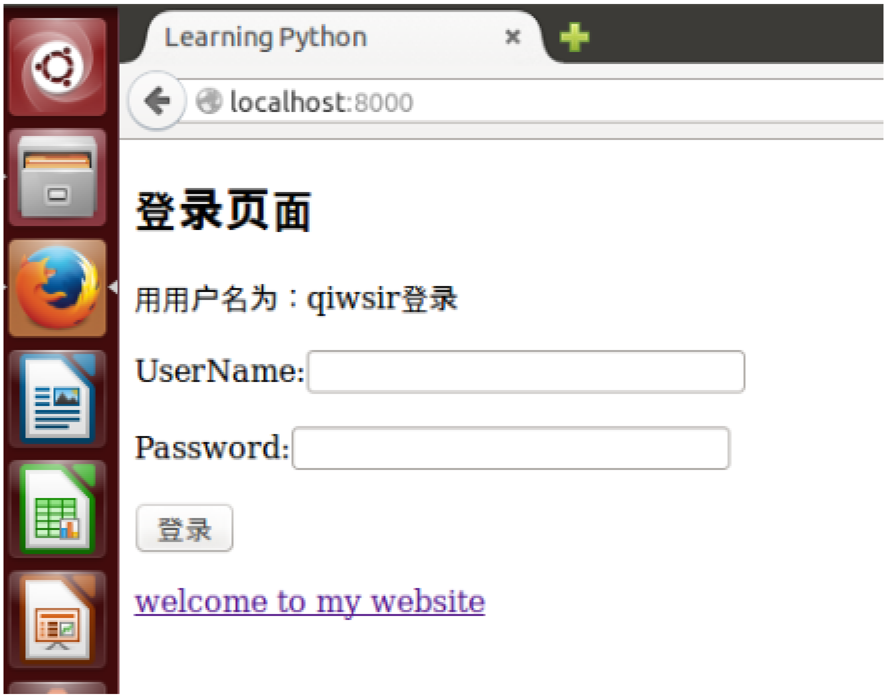
登录

welcome to my website

[illegible] `{{ website }}` [illegible]

```
{% raw website %}
```

[illegible]



Learning Python

localhost:8000

登录页面

用用户名为：qiwsir登录

UserName:

Password:

登录

welcome to my website

□□□□□□□□□□□□□□

```
{% autoescape None %}
{{ website }}
```

□□□□□□□□□□□□□

□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

- escape(s)□□□□□□ s □□ &□<□> □□□□□□ HTML □□□
- url_escape(s)□□□ urllib.quote_plus □□□□□ s □□□□□ URL □□□□□
- json_encode(val)□□ val □□□ JSON □□□
- squeeze(s)□□□□□□ s□□□□□□□□□□□□□□□□□□□□

□□□

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

□ tornado □□□ (5)

□□□□

□□ Python □□□□□□□□□“□□”□□□□□□□□□□□□□□□□□□□ (4)□□□“□□”□□]□□□□□□□□□□□□□□□□□□

□ tornado □□□□□□□□□□

□□□□□□□□□□□ base.html□□□□□□

```
 <!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1" />
    <title>Learning Python</title>
</head>
<body>
    <header>
        {% block header %}{% end %}
    </header>
    <content>
        {% block body %}{% end %}
    </content>
    <footer>
        {% set website = "<a
href='http://www.itdiffer.com'>welcome to my
website</a>" %}
        {% raw website %}
    </footer>
    <script src="
{{static_url("js/jquery.min.js")}}"></script>
    <script src="
{{static_url("js/script.js")}}"></script>
</body>
</html>
```

□□□□□ base.html □□□□□□□□□□□□□ index.html □ user.html □
□□

index.html □□□□□

```
 {% extends "base.html" %}

{% block header %}
    <h2>□□□□</h2>
    <p>□□□□□□{{user}}□□</p>
{% end %}
{% block body %}
    <form method="POST">
        <p><span>UserName:</span><input
type="text" id="username"/></p>
        <p><span>Password:</span><input
type="password" id="password" /></p>
        <p><input type="BUTTON" value="□□"
id="login" /></p>
    </form>
{% end %}
```

user.html □□□□□□

```
 {% extends "base.html" %}
```

```
{% block header %}
    <h2>Your informations are:</h2>
{% end %}

{% block body %}
    <ul>
        {% for one in users %}
            <li>username:{{one[1]}}</li>
            <li>password:{{one[2]}}</li>
            <li>email:{{one[3]}}</li>
        {% end %}
    </ul>
{% end %}
```

[illegible] `{% extends "base.html" %}` [illegible] base.html [illegible] base.html [illegible] `{% block header %}{% end %}` [illegible] index.html [illegible] user.html [illegible] base.html [illegible]

CSS

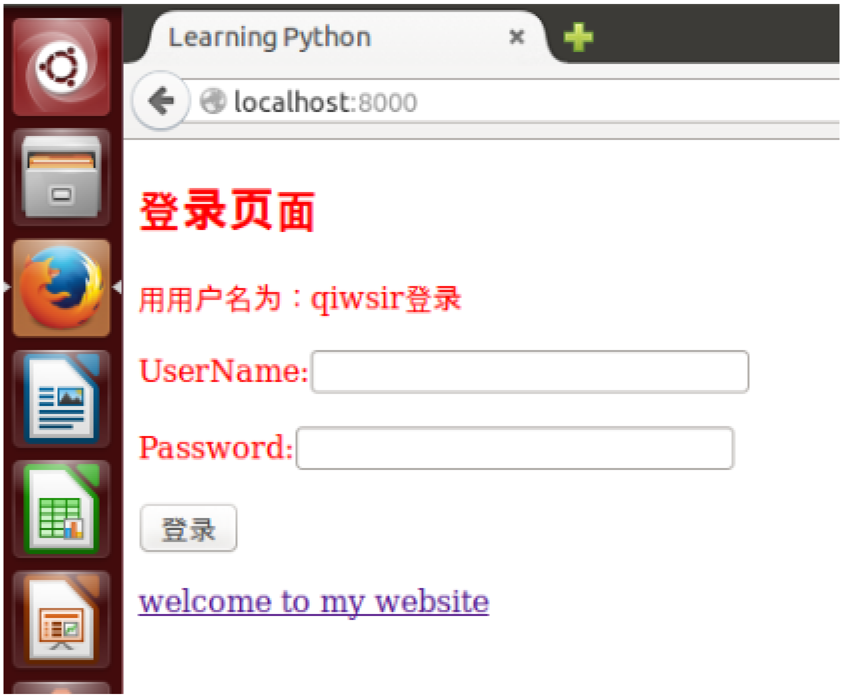
[illegible] css[illegible] is [illegible] base.html [illegible] `<head></head>` [illegible]

```
<link rel="stylesheet" type="text/css" href="{{static_url("css/style.css")}}">
```

[illegible] style.css [illegible]

```
body {
    color:red;
}
```

[illegible]



Learning Python
localhost:8000

登录页面

用用户名为：qiwsir登录

UserName:

Password:

登录

welcome to my website

[illegible]

[illegible] CSS [illegible] CSS [illegible]

[illegible] tornado [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]

cookie [illegible]

cookie [illegible] cookie[illegible]

> Cookie[illegible] Cookies[illegible]Client Side[illegible] RFC2109[illegible] Lou Montulli [illegible] 1993 [illegible] 3 [illegible]

[illegible] cookie [illegible]

> [illegible] HTTP [illegible] Web [illegible] HTTP [illegible] Cookie [illegible] HTTP [illegible]“[illegible]”[illegible] Cookies [illegible]
>
> [illegible] Cookie[illegible] Cookie [illegible] Cookie [illegible] Cookie [illegible]
>
> Cookie [illegible]“[illegible]”[illegible] Cookie [illegible] Cookie [illegible] Cookie[illegible]

[illegible]cookie [illegible]

1. cookie [illegible] HTTP [illegible]
2. [illegible] HTTP [illegible] cookie [illegible] HTTPS[illegible]
3. Cookie [illegible] 4KB [illegible]

[illegible] cookie[illegible] cookie[illegible] cookie [illegible]

Cookie [illegible] cookie [illegible] cookie[illegible] cookie [illegible] cookie [illegible]“[illegible](Cross site scripting)”[illegible]

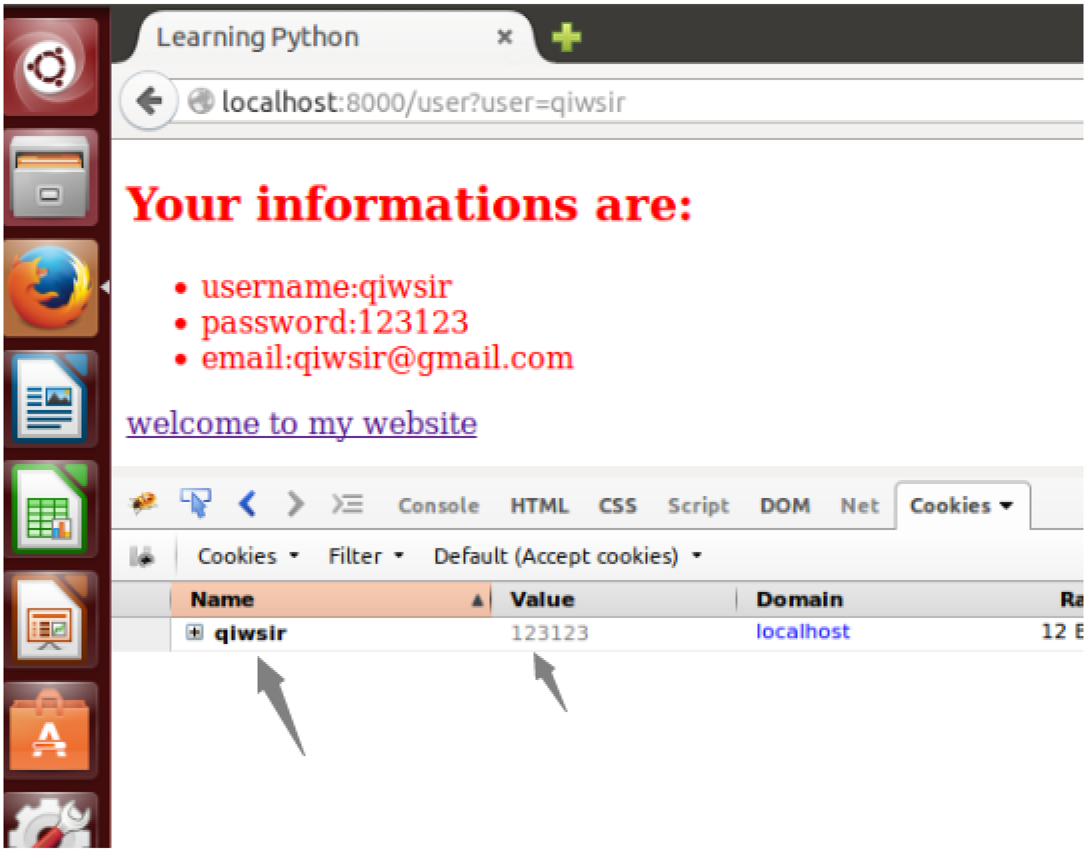
> [illegible]Cross-site scripting[illegible] XSS [illegible] HTML [illegible]
>
> XSS [illegible] JavaScript[illegible] Java[illegible] VBScript[illegible] ActiveX[illegible] Flash [illegible] HTML[illegible] cookie [illegible]

cookie [illegible] tornado [illegible] cookie [illegible]

`set_cookie()` [illegible] `get_cookie()` [illegible]

[illegible] index.py [illegible] IndexHandler [illegible] post() [illegible] cookie[illegible] def post(self): username = self.get_argument("username") password = self.get_argument("password") user_infos = mrd.select_table(table="users",column="*",condition="username",value=username) if user_infos: db_pwd = user_infos[0][2] if db_pwd == password: self.set_cookie(username,db_pwd) #[illegible] cookie self.write(username) else: self.write("your password was not right.") else: self.write("There is no thi user.")

[illegible] `self.set_cookie(username,db_pwd)`[illegible]


Learning Python
localhost:8000/user?user=qiwsir
Console
HTML
CSS
Script
DOM
Net
Cookies
Cookies
Filter
Default (Accept cookies)

Your informations are:

- username:qiwsir
- password:123123
- email:qiwsir@gmail.com

welcome to my website

Name	Value	Domain
qiwsir	123123	localhost

□□
□□□□ cookie □□□□

□□□□□□□□□□□

tornado □□□□□□□□□□□□set_secure_cookie() □ get_secure_cookie()□□□□□□ cookie□□□□□□□□□□□□□□□□□□□ set_cookie() □□□□□□□ set_secure_cookie() □□□□ cookie □□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□cookie □□□□□□

□□□ set_secure_cookie() □□□□ cookie□□□□ application.py □ □□ setting □□□□□□□□

```
setting = dict(
    template_path =
os.path.join(os.path.dirname(__file__),
"templates"),
    static_path =
os.path.join(os.path.dirname(__file__),
"statics"),
    cookie_secret =
"bZJc2sWbQLKos6GkHn/VB9oXwQt8S0R0kRvJ5/xJ89E=",
    )
```

□□ `cookie_secret = "bZJc2sWbQLKos6GkHn/VB9oXwQt8S0R0kRvJ5/xJ89E="` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□ tornado □□ cookie □□□□ Base-64 □□□□□□□□□□□□□□□ cookie □□□ HMAC □□□□□□cookie_secret □□□□□□□□□□□□□□□□□□□□□□□□□□□

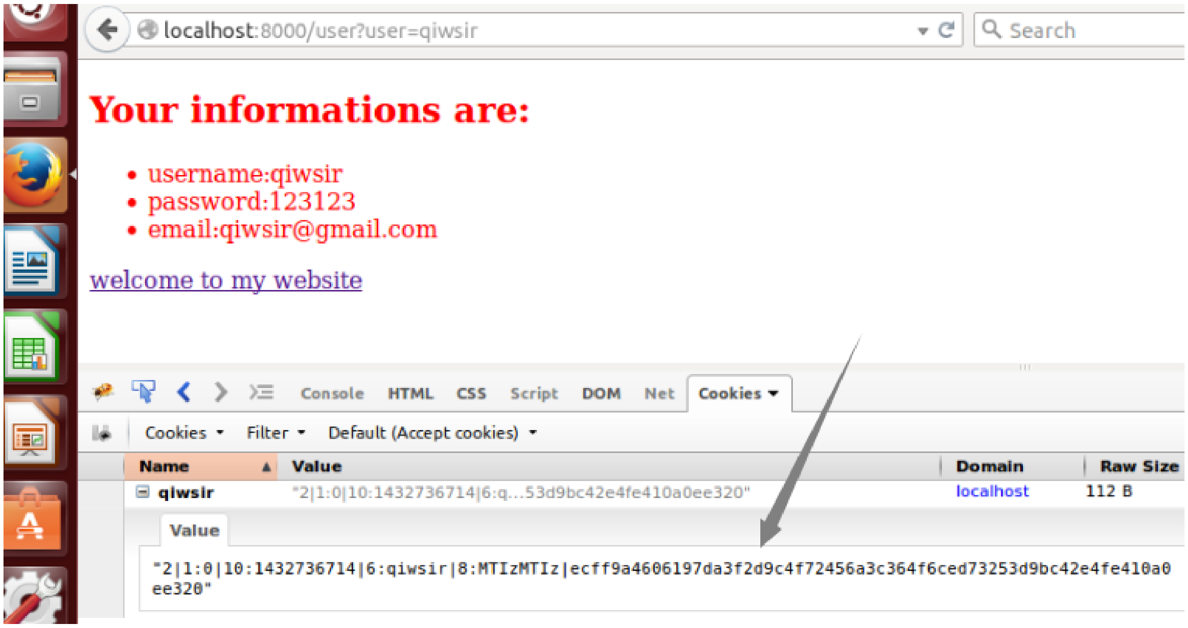
```
>>> import base64, uuid
>>> base64.b64encode(uuid.uuid4().bytes)
'w8yZud+kRHiP9uABEXaQiA=='
```

□□□□□□□□□□□□□□ `base64.b64encode(uuid.uuid4().bytes + uuid.uuid4().bytes)` □□□□□□□□□□□□□□□□□□□□□□ cookie_secret □□

□□□□ index.py □□□ cookie □□□□□□□

```
self.set_secure_cookie(username,db_pwd)
```

□□□□□□□□□□□


localhost:8000/user?user=qiwsir

Your informations are:

- username:qiwsir
- password:123123
- email:qiwsir@gmail.com

welcome to my website

Search
Console
HTML
CSS
Script
DOM
Net
Cookies
Cookies
Filter
Default (Accept cookies)

Name	Value	Domain	Raw Size
qiwsir	"2|1:0|10:1432736714|6:q...53d9bc42e4fe410a0ee320"	localhost	112 B

Value

```
"2|1:0|10:1432736714|6:qiwsir|8:MTIzMTIz|ecff9a4606197da3f2d9c4f72456a3c364f6ced73253d9bc42e4fe410a0
ee320"
```

□□□□□“□”□□□□

□□□□□□ cookie□□ `self.get_secure_cookie(username)` □□□

□□ cookie □□□□□□□□□□□□□□□□ httponly □ secure □□□□□□□ cookie □□□□□□□□□

```
self.set secure cookie(username, db_pwd,
httponly=True, secure=True)
```

□□□ cookie□□□□□ `self.set_secure_cookie(username)` □□□□□□□□ user.py □□□□□□□□□□□□□□□ print □□□□□□□□□□□□□□ username □□□ cookie □□□□□□□□□□□□“□”□□□□□□□□□

□□□□□□□□□□□□ SSL □□□□ cookie□□□□□□□□□□□□□□□□□□

XSRF

XSRF [illegible] Cross-site request forgery[illegible]"one click attack"[illegible] CSRF [illegible] XSRF[illegible]"sea surf"[illegible]XSS[illegible]XSS [illegible] XSRF [illegible] XSS [illegible]XSRF [illegible] [illegible] XSS [illegible]

[illegible] XSRF[illegible]CSRF | XSRF [illegible]

[illegible] XSRF [illegible] HTTP [illegible] post() [illegible] RESTful [illegible]

> [illegible]REST[illegible] web [illegible]
>
> [illegible]Representational State Transfer[illegible] REST[illegible] Roy Fielding [illegible] 2000 [illegible] Web [illegible] REST [illegible] SOAP [illegible] XML-RPC [illegible] web [illegible] REST [illegible]Amazon.com [illegible] REST [illegible] Web [illegible] Web [illegible] REST [illegible]
>
> [illegible]

[illegible] tornado [illegible] XSRF [illegible]

[illegible] application.py [illegible] xsrf_cookies [illegible] XSRF [illegible]

```
setting = dict(
    template_path =
os.path.join(os.path.dirname(__file__),
"templates"),
    static_path =
os.path.join(os.path.dirname(__file__),
"statics"),
    cookie_secret =
"bZJc2sWbQLKos6GkHn/VB9oXwQt8S0R0kRvJ5/xJ89E=",
    xsrf_cookies = True,
)
```

□□□□□□□Tornado □□□□□□□□□□□□□□`_xsrf` □□ post/put/delete □□□tornado □□□□□□□□□`_xsrf` cookies□□□□ □□□□□□□□ XSRF □□□□□□□□□□□□ index.html □□□□□□□□□

```
 {% extends "base.html" %}

{% block header %}
    <h2>□□□□</h2>
    <p>□□□□□□{{user}}□□</p>
{% end %}
{% block body %}
    <form method="POST">
        {% raw xsrf form html() %}
        <p><span>UserName:</span><input
type="text" id="username"/></p>
        <p><span>Password:</span><input
type="password" id="password" /></p>
        <p><input type="BUTTON" value="□□"
id="login" /></p>
    </form>
{% end %}
```

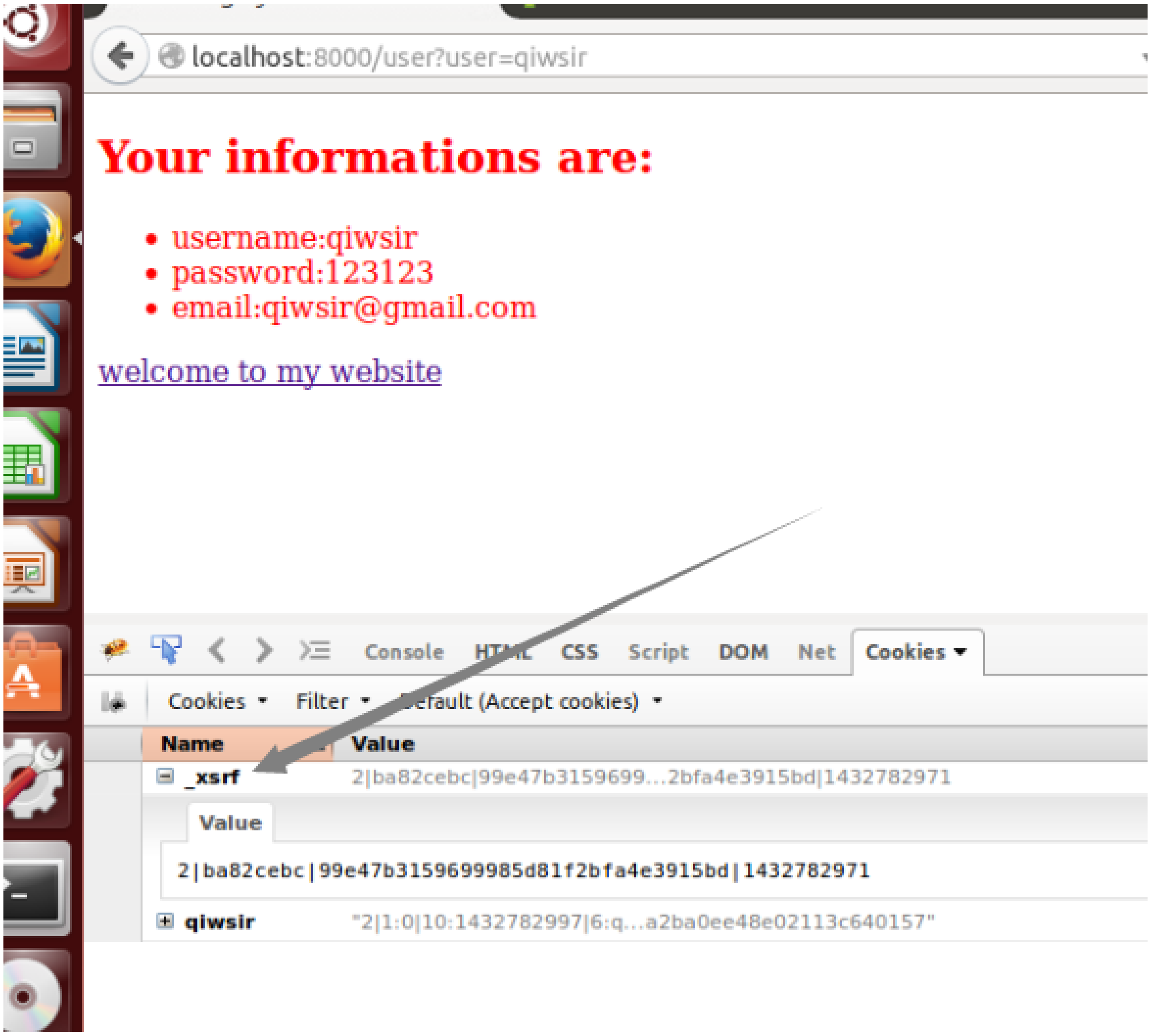
`{% raw xsrf_form_html() %}`□□□□□□□□□□□□□□□□□□□□□□ □□□□□□

□□□□□□□□□□□□□ ajax()□□□□□ ajax □□□□□□□□ _xsrf □□□

□□□ script.js □□□

```
  function getCookie(name){
    var x = document.cookie.match("\\b" + name +
"=([^;]*)\\b");
    return x ? x[1]:undefined;
}

$(document).ready(function(){
    $("#login").click(function(){
        var user = $("#username").val();
        var pwd = $("#password").val();
        var pd = {"username":user,
"password":pwd, "_xsrf":getCookie("_xsrf")};
        $.ajax({
            type:"post",
            url:"/",
            data:pd,
            cache:false,
            success:function(data){
                window.location.href = "/user?
user="+data;
            },
            error:function(){
                alert("error!");
            },
        });
    });
});
```

□□ getCookie() □□□□□□ cookie □□□□□□□□□□□□□ post □□□□ `var pd = {"username":user, "password":pwd, "_xsrf":getCookie("_xsrf")};` □□□□□□□


localhost:8000/user?user=qiwsir

Your informations are:

- username:qiwsir
- password:123123
- email:qiwsir@gmail.com

welcome to my website

Console
HTML
CSS
Script
DOM
Net
Cookies
Cookies
Filter
Default (Accept cookies)

Name	Value
_xsrf	2|ba82cebc|99e47b3159699...2bfa4e3915bd|1432782971
qiwsir	"2|1:0|10:1432782997|6:q...a2ba0ee48e02113c640157"

Value

2|ba82cebc|99e47b3159699985d81f2bfa4e3915bd|1432782971

□□ tornado □□□ XSRF □□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□ □□□□□□□□□□□□□ □□□□□□□WebAppSec/Secure Coding Guidelines

□□□ □□

□□□

□□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

[illegible] tornado [illegible] (6)

[illegible]

[illegible]

[illegible] cookie [illegible] tornado [illegible]

[illegible] tornado [illegible]

[illegible] handler [illegible] base.py[illegible]

```
#! /usr/bin/env python
# coding=utf-8

import tornado.web

class BaseHandler(tornado.web.RequestHandler):
    def get_current_user(self):
        return self.get_secure_cookie("user")
```

[illegible] BaseHandler [illegible] cookie[illegible]

□□□□□□□□□□□□□□□□□□□`__init__()`□□□□□□□□□□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□ index.py □□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.escape
import methods.readdb as mrd
from base import BaseHandler

class IndexHandler(BaseHandler):    #□□ base.py
□□□ BaseHandler
    def get(self):
    usernames =
mrd.select_columns(table="users",column="usernam
e")
    one user = usernames[0][0]
    self.render("index.html", user=one_user)

    def post(self):
        username = self.get argument("username")
        password = self.get_argument("password")
        user infos =
mrd.select table(table="users",column="*",condit
ion="username",value=username)
        if user infos:
            db pwd = user infos[0][2]
            if db pwd == password:
                self.set_current_user(username)
#□□□□□□□□ cookie□□□□□□
                self.write(username)
            else:
                self.write("-1")
        else:
            self.write("-1")

    def set current_user(self, user):
        if user:
            self.set secure cookie('user',
tornado.escape.json encode(user))    #□□□□□□□
tornado.escape.json_encode() □□
        else:
            self.clear_cookie("user")

class ErrorHandler(BaseHandler):    #□□□□□□□□□□□
□□□□□
    def get(self):
#□□□□□□□□□□□□□□□□□□□□
        self.render("error.html")
```

□ index.py □□ IndexHandler □□□□□ BaseHandler □□□□□□□□□□□ set_current_user() □□□□□□□□ cookie□□□□□□□□□□ tornado.escape.json_encode() □□□□□□□□

> tornado.escape.json_encode(value) JSON-encodes the given Python object.

□□□□□□□□□□□□□
http://www.tornadoweb.org/en/branch2.3/escape.html

□□□□□□□□ user □□□ json□□□□□ cookie □□□□□ cookie □□□□□□□□□ user □□□□□□□□□

> tornado.escape.json_decode(value) Returns Python objects for the given JSON string

□□□json □□□□ dump()□load()□□□□□

□□□□□ user.py □□□□□□□□

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.web
import tornado.escape
import methods.readdb as mrd
from base import BaseHandler

class UserHandler(BaseHandler):
    @tornado.web.authenticated
    def get(self):
        #username = self.get_argument("user")
        username = 
tornado.escape.json_decode(self.current_user)
        user_infos = 
mrd.select_table(table="users",column="*",condit
ion="username",value=username)
        self.render("user.html", users = 
user_infos)
```

[illegible] get() [illegible] `@tornado.web.authenticated` [illegible] tornado [illegible] `username = self.get_argument("user")` [illegible] url [illegible] `self.current_user` [illegible] setting [illegible] login_url [illegible] setting [illegible]

[illegible] index.py [illegible] set_current_user() [illegible] user [illegible] json [illegible] cookie [illegible] `username = tornado.escape.json_decode(self.current_user)` [illegible] username [illegible]

application.py [illegible] setting [illegible]

```
#!/usr/bin/env Python
# coding=utf-8
```

```
from url import url

import tornado.web
import os

setting = dict(
    template_path =
os.path.join(os.path.dirname(__file__),
"templates"),
    static_path =
os.path.join(os.path.dirname(__file__),
"statics"),
    cookie_secret =
"bZJc2sWbQLKos6GkHn/VB9oXwQt8S0R0kRvJ5/xJ89E=",
    xsrf_cookies = True,
    login_url = '/',
)

application = tornado.web.Application(
    handlers = url,
    **setting
)
```

[illegible] `login_url = '/',` [illegible] `/login` [illegible] `login_url = '/login'` [illegible]

[illegible]

[illegible]

[illegible]

[illegible]**qiwsir@126.com**,[illegible]

[illegible] tornado [illegible] (7)

[illegible] tornado[illegible] c10k [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

- [illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]
- [illegible]“[illegible] Python”[illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

> [illegible] (synchronous communication/ asynchronous communication)

> [illegible]“[illegible]”[illegible]“[illegible]”[illegible] [illegible]“[illegible]”[illegible]“[illegible]”[illegible]
>
> [illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

[illegible]

[illegible]

“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible][illegible][illegible]“[illegible]”[illegible]

> [illegible].
>
> [illegible]

[illegible]“[illegible] PHP [illegible]”[illegible]

[illegible] http://www.cppblog.com/converse/archive/2009/05/13/82879.html[illegible]

tornado [illegible]

[illegible] tornado [illegible] web[illegible]

[illegible] handlers [illegible] sleep.py

```
#!/usr/bin/env python
# coding=utf-8

from base import BaseHandler

import time

class SleepHandler(BaseHandler):
    def get(self):
        time.sleep(17)
        self.render("sleep.html")

class SeeHandler(BaseHandler):
    def get(self):
        self.render("see.html")
```

[illegible] tornado [illegible] (1)[illegible]

sleep.html [illegible] see.html [illegible] url.py [illegible]

[illegible] `localhost:8000/sleep` [illegible] 1[illegible] `localhost:8000/see` [illegible] 2[illegible] 8000 [illegible]“ctrl-tab”[illegible]

1. [illegible] 1[illegible]
2. [illegible] 2[illegible]
3. [illegible]
4. [illegible] 1 [illegible] 2 [illegible]

[illegible] sleep.py [illegible] time.sleep(17) [illegible]

[illegible]

[illegible]

tornado [illegible] tornado [illegible] tornado.ioloop.IOLoop[illegible] get() [illegible] `time.sleep(17)` [illegible] tornado [illegible]

[illegible]tornado [illegible] `@tornado.web.asynchronous`[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.web
from base import BaseHandler

import time

class SleepHandler(BaseHandler):
    @tornado.web.asynchronous
    def get(self):

tornado.ioloop.IOLoop.instance().add timeout(tim
e.time() + 17, callback=self.on_response)
    def on response(self):
        self.render("sleep.html")
        self.finish()
```

[illegible] sleep.py [illegible] get() [illegible] `@tornado.web.asynchronous` [illegible] tornado [illegible] `_auto_fininsh` [illegible] false[illegible] get() [illegible] `@tornado.web.asynchronous` [illegible] `self.finish()` [illegible]

`tornado.ioloop.IOLoop.instance().add_timeout()` [illegible] `time.time()+17` [illegible] `time.sleep(17)` [illegible] `on_response()` [illegible] `self.render("sleep.html")` [illegible] `self.finish()`[illegible]

[illegible] `time.sleep(17)` [illegible]

[illegible] `tornado.ioloop.IOLoop.instance().add_timeout(time.time() + 17, callback=self.on_response)` [illegible] `callback=self.on_response` [illegible] `self.render("sleep.html")` [illegible] on_response(self) [illegible] self.render("sleep.html") [illegible] self.finish()` [illegible]

[illegible]“[illegible]”[illegible]“[illegible]”[illegible]“[illegible]”[illegible]

```
#!/usr/bin/env Python
# coding=utf-8

import tornado.web
import tornado.gen
from base import BaseHandler
```

```
import time

class SleepHandler(tornado.web.RequestHandler):
    @tornado.gen.coroutine
    def get(self):
        yield
tornado.gen.Task(tornado.ioloop.IOLoop.instance(
).add_timeout, time.time() + 17)
        #yield tornado.gen.sleep(17)
        self.render("sleep.html")
```

□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□

□□□□□□ `@tornado.gen.coroutine` □□□□□□□□□□□ `import tornado.gen` □□□□□□□□□□□ `@tornado.gen.engine` □□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□

> This decorator(□ engine) is similar to coroutine, except it does not return a Future and the callback argument is not treated specially.

`@tornado.gen.engine` □□□□□□□□□□□□□□ `@tornado.gen.corroutine` □□□□□□ tornado 3.0 □□□□□□□□ □□□□□□□□□□□□□□□ `@tornado.gen.engine` □□□□□□□□□□□□□ □□□□□□□□□□□□□□ `@tornado.gen.coroutine` □□□□□□□□□□□ □□□□□□□□□□□□□□□□

□□□□□ get() □□□□□ yield □□□□□□□□□□□□□□□□□□□□□□ `yield tornado.gen.Task(tornado.ioloop.IOLoop.instance().add_timeout, time.time() + 17)` □□□□□□□□□□□□□□□□□ □□□□□□□□□ `time.sleep(17)` □□□□□ `tornado.gen.Task()` □□□□□□□“Adapts a callback-based asynchronous function for use in coroutines.”□□□□□□□□□□□□□□□□□□□□□□□□□□ yield □□□

[illegible] `yield tornado.gen.sleep(17)` [illegible] `tornado.gen.Task()` [illegible] `yield tornado.gen.sleep()` [illegible]

[illegible] tornado [illegible] tornado [illegible]

[illegible]

[illegible] tornado [illegible] tornado [illegible]

- [illegible] SQL [illegible] noSQL[illegible]connect[illegible]insert[illegible] update [illegible]
- [illegible]
- time.sleep[illegible]
- smtplib[illegible]
- [illegible] tornado [illegible] httpclient [illegible] pycurl [illegible]

[illegible]

[illegible]/[illegible]

- [illegible] mysql[illegible]adb[illegible] python [illegible] mysql [illegible] mongodb [illegible] tornado [illegible] mongodb [illegible] motor[illegible] logo[illegible] http://motor.readthedocs.org/en/stable/[illegible]

- [illegible] IO[illegible]Redis[illegible]MongoDB[illegible]
- time.sleep() [illegible] tornado [illegible] `tornado.gen.sleep()` [illegible] `tornado.ioloop.IOLoop.instance().add_timeout` [illegible]
- smtp [illegible] tornado-smtp-client[illegible]
- [illegible] tornado.httpclient.AsyncHTTPClient[illegible]

[illegible]Async Client Libraries built on tornado.ioloop[illegible]

[illegible] mongodb [illegible] motor [illegible] tornado [illegible]python[illegible]greenlet-tornado[illegible]

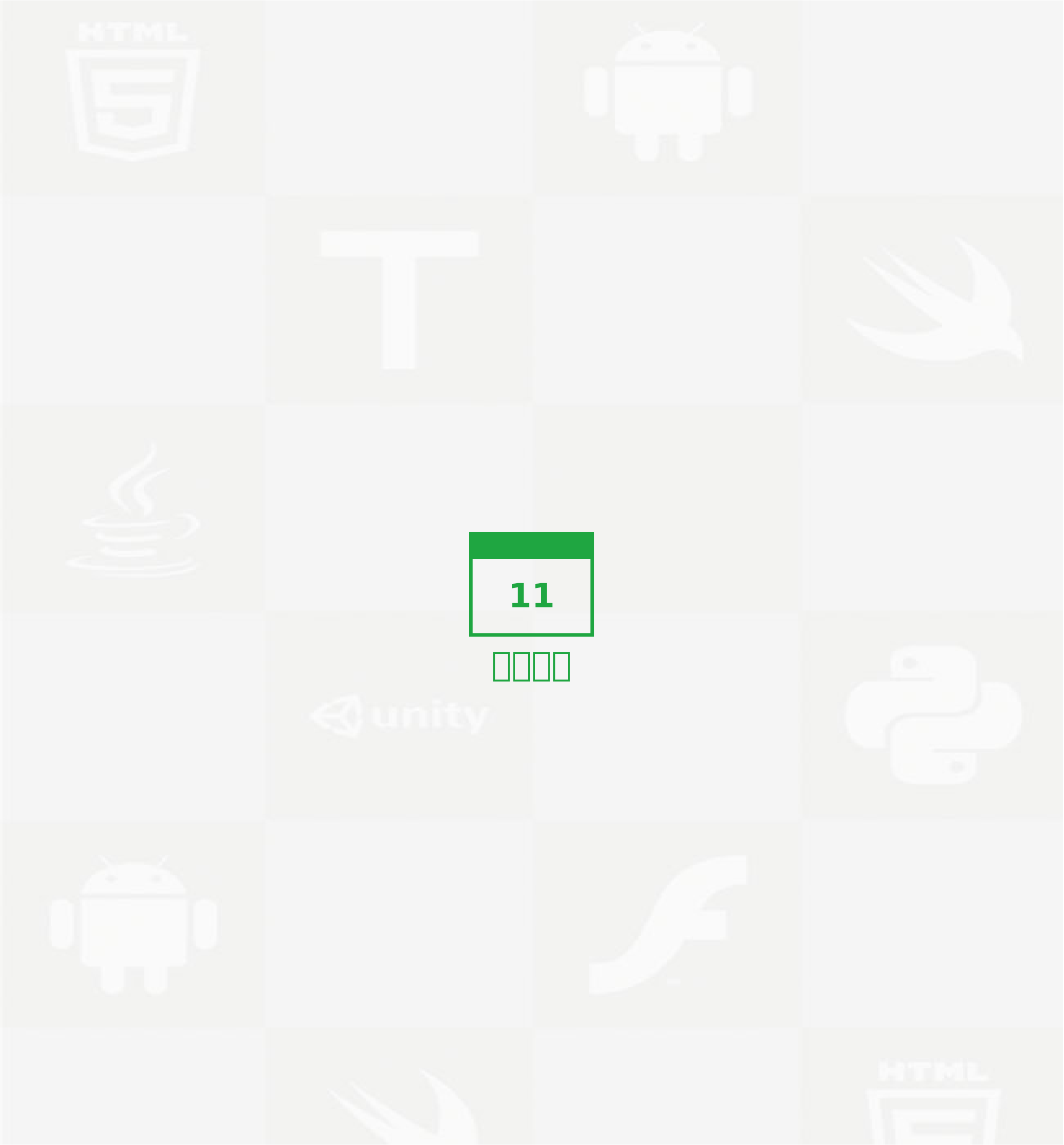
[illegible] mysql [illegible] mysql [illegible]

[illegible] tornado [illegible]

[illegible]http://emptysqua.re/blog/refactoring-tornado-coroutines/

[illegible]

[illegible]**qiwsir@126.com**,[illegible]



11

[illegible]

[illegible]

[illegible]

[illegible]

[illegible] Fortran[illegible] Fortran[illegible]

> Fortran [illegible]1953 [illegible] 12 [illegible]IBM [illegible]·[illegible]J. Backus[illegible]·[illegible]Cuthbert Hurd[illegible] IBM704 [illegible] IBM [illegible]·[illegible]1957 [illegible]IBM [illegible] FORTRAN [illegible] IBM704 [illegible] FORTRAN [illegible]1957 [illegible] 4 [illegible] 20 [illegible] IBM [illegible] FORTRAN [illegible]“[illegible]……[illegible]”[illegible] FORTRAN [illegible]1958 [illegible] FORTRAN Ⅱ[illegible] FORTRAN Ⅲ[illegible]1962 [illegible] FORTRAN Ⅳ [illegible] Fortran 2008[illegible]

[illegible] matlab[illegible]

> MATLAB[illegible] MATrix LABoratory [illegible] The MathWorks [illegible]MATLAB [illegible]/[illegible]MATLAB [illegible] C,C++,Java,Python [illegible] FORTRAN[illegible]

[illegible]

[illegible] R [illegible]

> R [illegible]R [illegible] Ross Ihaka [illegible] Robert Gentleman [illegible] R[illegible]“R [illegible]”[illegible]R [illegible] S [illegible] GNU [illegible] S [illegible] S [illegible] R [illegible]R [illegible] Scheme[illegible]

[illegible] Python[illegible] Python [illegible] Python [illegible] Python [illegible] Python[illegible]

- [illegible]
- [illegible]
- [illegible] Python [illegible]
- [illegible] Python[illegible] web [illegible]

[illegible] Python [illegible] Fortran [illegible] R[illegible] Python[illegible]

[illegible]

[illegible] Ubuntu [illegible] Debian[illegible]

```
sudo apt-get install Python-numpy Python-scipy Python-matplotlib ipython ipython-notebook Python-pandas Python-sympy Python-nose
```

□□

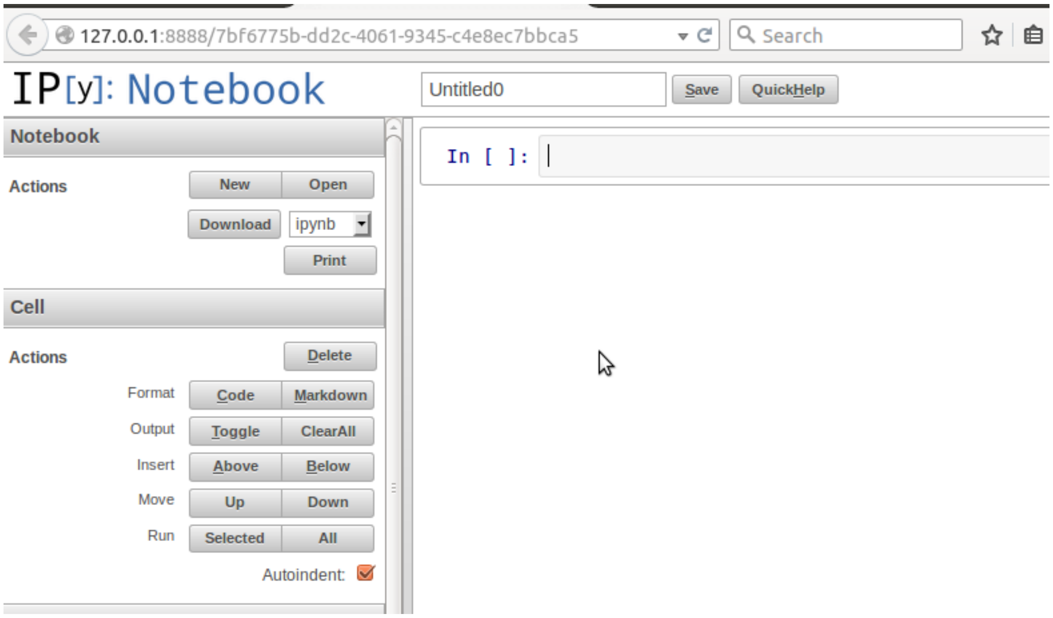
□□□□□□□□□□□□ windows □□□ http://www.scipy.org/install.html

□□□□

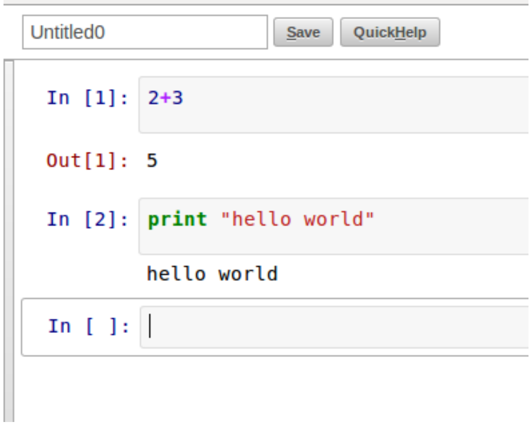
□□□□□□□□□□□□□□□□□ ipython notebook□□□□□□□□□□ shell □□□

```
ipython notebook --pylab=inline
```

□□□□□□□□□□□□□□□□□□□□□

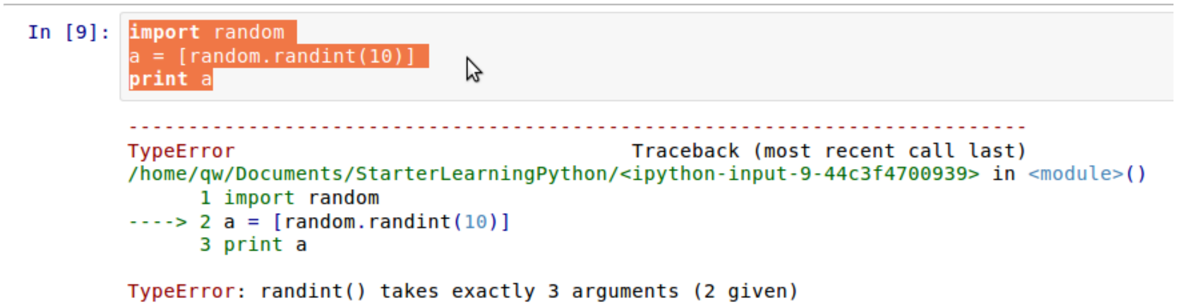

127.0.0.1:8888/7bf6775b-dd2c-4061-9345-c4e8ec7bbca5
Search
IP[y]: Notebook
Untitled0
Save
QuickHelp
Notebook
Actions
New
Open
Download
ipynb
Print
Cell
Actions
Delete
Format
Code
Markdown
Output
Toggle
ClearAll
Insert
Above
Below
Move
Up
Down
Run
Selected
All
Autoindent:
In [ ]:

□ In □□□□□□□□□□□ Python □□□□□□□□ `SHIFT+ENTER` □□ `CTRL+ENTER` □□□□□□□□□□□ `ENTER`□□□□ □□□□□□□□□□□□□


Untitled0
Save
QuickHelp

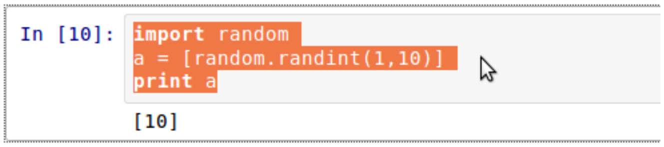
```
In [1]: 2+3

Out[1]: 5

In [2]: print "hello world"

        hello world

In [ ]:
```

Ipython Notebook □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□



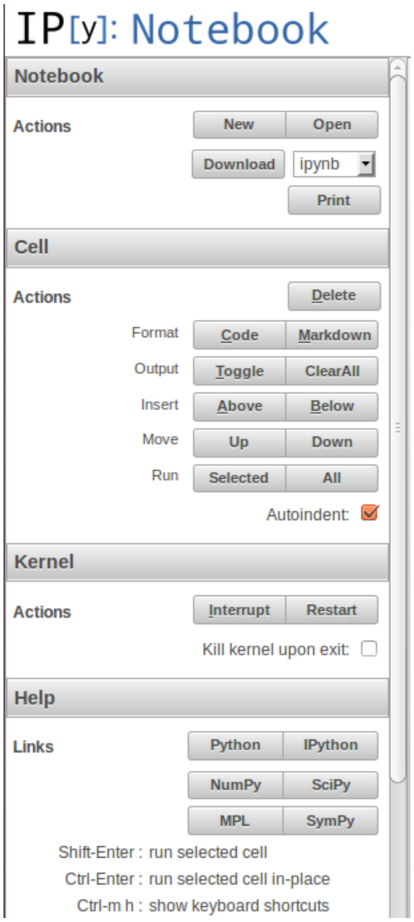
```
In [9]: import random
        a = [random.randint(10)]
        print a

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/home/qw/Documents/StarterLearningPython/<ipython-input-9-44c3f4700939> in <module>()
      1 import random
----> 2 a = [random.randint(10)]
      3 print a

TypeError: randint() takes exactly 3 arguments (2 given)
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□



```
In [10]: import random
         a = [random.randint(1,10)]
         print a
         [10]
```

□□□□□□□□□□□□□□□ ipython notebook □□□□□□


IP[y]: Notebook
Notebook
Actions
New
Open
Download
ipynb
Print
Cell
Actions
Delete
Format
Code
Markdown
Output
Toggle
ClearAll
Insert
Above
Below
Move
Up
Down
Run
Selected
All
Autoindent:
Kernel
Actions
Interrupt
Restart
Kill kernel upon exit:
Help
Links
Python
IPython
NumPy
SciPy
MPL
SymPy

Shift-Enter : run selected cell
Ctrl-Enter : run selected cell in-place
Ctrl-m h : show keyboard shortcuts

□□□□□□□□□□□□□□ Python □□□□□□□□□□□□□□□ shell □□□□□□□□□□□

```
$ ipython
```

□□□□□□□□ Python □□□□□□□□□□□□

```
In [1]: print "hello, pandas"
hello, pandas

In [2]:
```

□□□ ipython □□□□□□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

Pandas □□ (1)

Pandas □□□ NumPy □□□

□□□□□□□

Pandas □□□ Python □□□□□□□□Python □□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□Pandas □□□□□□□□□□□□□Series □ DataFrame□□□□□□□ □□□□□□

□□□□□□□□□

```
In [1]: from pandas import Series, DataFrame
        import pandas as pd
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□ ipython notebook□□□□□□□□□□□□□

Series

Series □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□[9, 3, 8]□□□□□□□□□□□□□□□

data	9	3	8
index	0	1	2

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

index	data
0	9
1	3
2	8

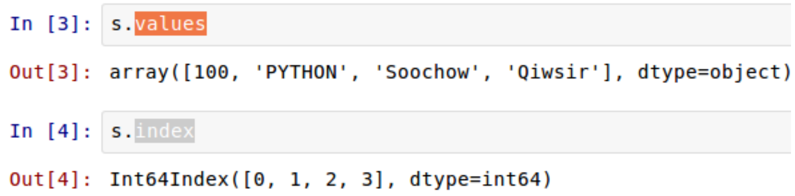
□□□□□□□□□□□□□□□□□□

Series □□“□□□”□ list□

```
In [2]: s = Series([100, "PYTHON", "Soochow", "Qiwsir"])
        s

Out[2]: 0       100
        1    PYTHON
        2   Soochow
        3    Qiwsir
```

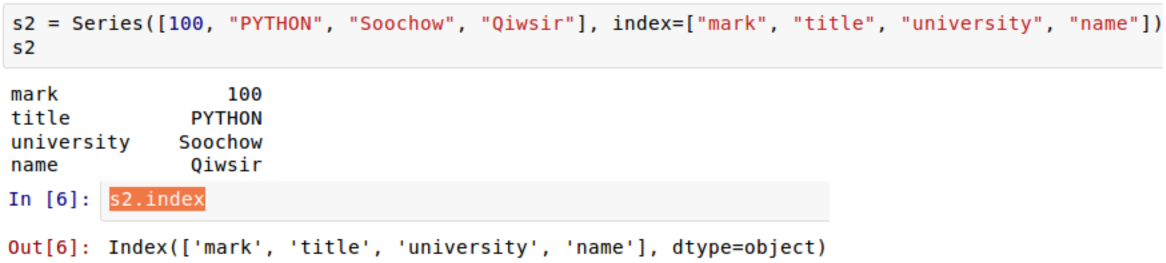
[illegible]

[illegible] Series [illegible] Series [illegible]



```
In [3]: s.values
Out[3]: array([100, 'PYTHON', 'Soochow', 'Qiwsir'], dtype=object)

In [4]: s.index
Out[4]: Int64Index([0, 1, 2, 3], dtype=int64)
```

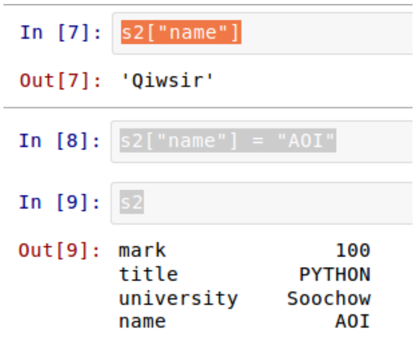
[illegible] 0 [illegible]Series [illegible]Series [illegible]



```
s2 = Series([100, "PYTHON", "Soochow", "Qiwsir"], index=["mark", "title", "university", "name"])
s2

mark              100
title          PYTHON
university    Soochow
name           Qiwsir
In [6]: s2.index

Out[6]: Index(['mark', 'title', 'university', 'name'], dtype=object)
```

[illegible]

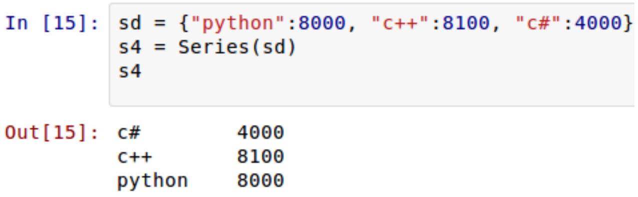
[illegible] list [illegible]Series [illegible]



```
In [7]: s2["name"]

Out[7]: 'Qiwsir'

In [8]: s2["name"] = "AOI"

In [9]: s2

Out[9]: mark              100
        title          PYTHON
        university    Soochow
        name              AOI
```

[illegible] dict [illegible]

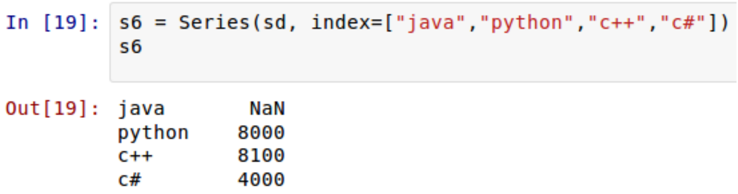
[illegible] Series [illegible] Series() [illegible] index[illegible] Series [illegible]



```
In [15]: sd = {"python":8000, "c++":8100, "c#":4000}
         s4 = Series(sd)
         s4

Out[15]: c#        4000
         c++       8100
         python    8000
```

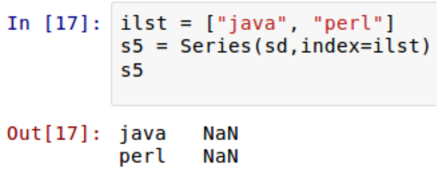
[illegible] dict [illegible]

[illegible]Pandas [illegible]“[illegible]”[illegible]



```
In [19]: s6 = Series(sd, index=["java","python","c++","c#"])
         s6

Out[19]: java       NaN
         python    8000
         c++       8100
         c#        4000
```

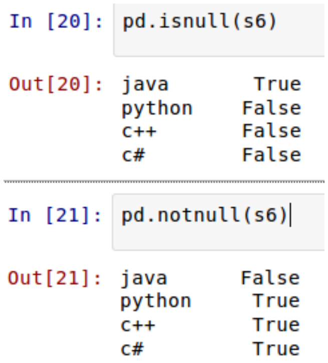
□ sd □□□□ `'python':8000, 'c++':8100, 'c#':4000` □□□"java"□□□□□□□□□□□□□□□□□□“□□□”□□□□□□□□□□□"java"□□□□□ Series □□□□□□□□□□□□□□□□□ `NaN` □□ Pandas □□□□□□□□□□□□□ `NaN` □□□□□□□□□



```
In [17]: ilst = ["java", "perl"]
         s5 = Series(sd,index=ilst)
         s5

Out[17]: java   NaN
         perl   NaN
```

□□□□ Series □□□□□ sd □□□□□□□□□□□□□ `NaN` □

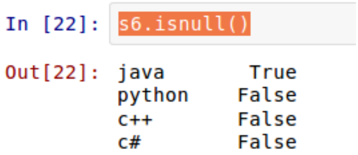
Pandas □□□□□□□□□□□□□□□



```
In [20]: pd.isnull(s6)

Out[20]: java       True
         python    False
         c++       False
         c#        False
```

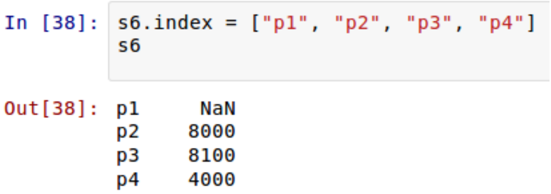
```
In [21]: pd.notnull(s6)

Out[21]: java      False
         python     True
         c++        True
         c#         True
```

□□□Series □□□□□□□□□□



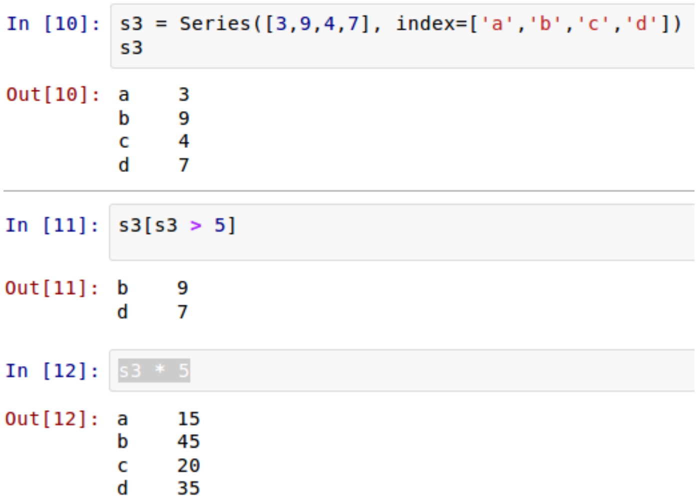
```
In [22]: s6.isnull()

Out[22]: java       True
         python    False
         c++       False
         c#        False
```

□□□□□□□□□□□□□□□□□□

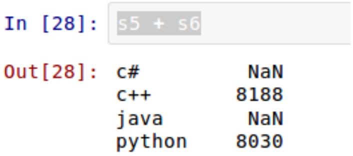


```
In [38]: s6.index = ["p1", "p2", "p3", "p4"]
         s6

Out[38]: p1     NaN
         p2    8000
         p3    8100
         p4    4000
```

□□ Series □□□□□□□□□□□□□□□□□□□□□□□□□□□□□



```
In [10]: s3 = Series([3,9,4,7], index=['a','b','c','d'])
         s3

Out[10]: a    3
         b    9
         c    4
         d    7

In [11]: s3[s3 > 5]

Out[11]: b    9
         d    7

In [12]: s3 * 5

Out[12]: a    15
         b    45
         c    20
         d    35
```

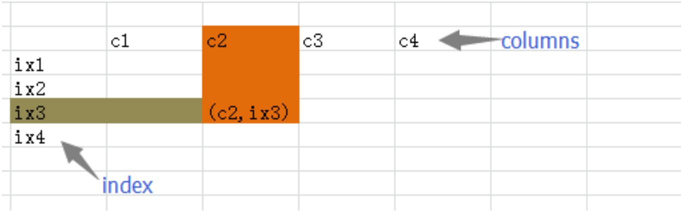


```
In [28]: s5 + s6

Out[28]: c#          NaN
         c++        8188
         java        NaN
         python     8030
```

□□□□□□□□□□□ ipython notebook □□□□□□□□□□□□□□□ Series □□□□□□□□□□ ipyton notebook□□□□□□ □□□□□□□□□□□□ ipython notebook □□□□□□□□□□□□□□□□ Python □□□□□□□□□

DataFrame

DataFrame □□□□□□□□□□□□□□□□□□□□□□□□□□ mysql □□□□□□□□□□□□□□□ columns□□□□□□□□ Series □□□□□□□ index□□□□□□□□□□ columns □ index □□□□□□□□□□□□□□□□ DataFrame □□□“□□□”□□□□□□□□ □□□“□”□□□□□□□□□□□)



□□□□□□□□□ Python □□□□□□□□□□□□□□□□ ipython notebook □□□□□□

```
 >>> import pandas as pd
>>> from pandas import Series, DataFrame

>>> data = {"name":["yahoo","google","facebook"], "marks":[200,400,800],
"price":[9, 3, 7]}
>>> f1 = DataFrame(data)
>>> f1
     marks  name      price
0    200    yahoo     9
1    400    google    3
2    800    facebook  7
```

□□□□□□ DataFrame □□□□□□□——□□ dict □□□□□□“□”□"name"□"marks"□"price"□□□ DataFrame □ columns □□□□□□□□□□□□□“□”□“□”□□□Series □□□□□□□□□□□□□ 0 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ excel □□ mysql □□□□□□□□

□□□□□□□□□columns □□□□□□□□□□□□□□□□□□□□□□□□ DataFrame □□columns □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
>>> f2 = DataFrame(data, columns=['name','price','marks'])
>>> f2
       name     price  marks
0     yahoo     9      200
1    google     3      400
2  facebook     7      800
```

□ Series □□□□DataFrame □□□□□□□□□□□□

```
>>> f3 = DataFrame(data, columns=['name', 'price', 'marks', 'debt'],
index=['a','b','c','d'])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib/pymodules/python2.7/pandas/core/frame.py", line 283, in
  init
    mgr = self. init dict(data, index, columns, dtype=dtype)
  File "/usr/lib/pymodules/python2.7/pandas/core/frame.py", line 368, in
 init dict
    mgr = BlockManager(blocks, axes)
  File "/usr/lib/pymodules/python2.7/pandas/core/internals.py", line 285,
in   init
    self. verify integrity()
  File "/usr/lib/pymodules/python2.7/pandas/core/internals.py", line 367,
in  verify integrity
    assert(block.values.shape[1:] == mgr_shape[1:])
AssertionError
```

□□ index □□——□□——□□□□□□□□□ data □□□□□□□□□□□□['a','b','c','d']□□

```
>>> f3 = DataFrame(data, columns=['name', 'price', 'marks', 'debt'],
index=['a','b','c'])
>>> f3
       name      price  marks  debt
a     yahoo      9      200     NaN
b    google      3      400     NaN
c  facebook      7      800     NaN
```

□□□□□□□□□□□□□□□□□□□□□ f3 □□□□columns □□□□□□□□□□□□('debt')□□□□□□ data □□□□□□□□□□□ debt □□□□□□□□□□□□ Pandas □□□□□ NaN □□□□□

□□ DataFrame □□□□□□□□□□□□□□□□□“□□□□□”□□□□

```
>>> newdata = {"lang":{"firstline":"python","secondline":"java"},
"price":{"firstline":8000}}
>>> f4 = DataFrame(newdata)
>>> f4
```

```
               lang     price
firstline      python   8000
secondline     java     NaN
```

[illegible]

```
 >>> DataFrame(newdata, index=["firstline","secondline","thirdline"])
               lang     price
firstline      python   8000
secondline     java     NaN
thirdline      NaN      NaN
```

[illegible] NaN[illegible]

[illegible] DataFrame [illegible] f3 [illegible] DataFrame[illegible]

```
 >>> f3.columns
Index(['name', 'price', 'marks', 'debt'], dtype=object)
```

DataFrame [illegible] columns [illegible] columns [illegible]

```
 >>> f3['name']
a       yahoo
b      google
c    facebook
Name: name
```

[illegible] Series[illegible] DataFrame [illegible] Series [illegible]

[illegible]

```
 >>> f3['debt'] = 89.2
>>> f3
       name     price  marks  debt
a     yahoo     9        200  89.2
b    google     3        400  89.2
c  facebook     7        800  89.2
```

[illegible]“[illegible]”[illegible] Series[illegible] DataFrame [illegible] Series [illegible] Series [illegible] DataFrame [illegible]

```
 >>> sdebt = Series([2.2, 3.3], index=["a","c"])    #[illegible]
>>> f3['debt'] = sdebt
```

[illegible] Series [illegible](sdebt [illegible]) [illegible] f3['debt'][illegible]Pandas [illegible]——[illegible]——[illegible] Series [illegible]"a","c"[illegible] DataFrame [illegible]

```
 >>> f3
       name  price  marks  debt
a     yahoo  9        200   2.2
b    google  3        400   NaN
c  facebook  7        800   3.3
```

□□□□□□□□□□□□□□□□□ NaN□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
 >>> f3["price"]["c"]= 300
>>> f3
       name   price   marks  debt
a     yahoo   9       200    2.2
b    google   3       400    NaN
c  facebook   300     800    3.3
```

□□□□□□□□□□□□□□□□ Pandas □□□□□□□□□

□□□

□□□□□□□□□□□□□□□□□□□**qiwsir@126.com**,□□□□□

Pandas □□ (2)

□□□□□□□□□□□□□□□□□□□□□□□ pandas □□□□□□□□□□□□“□□□□”□□ google □□□□

□□ csv □□

□□ csv □□

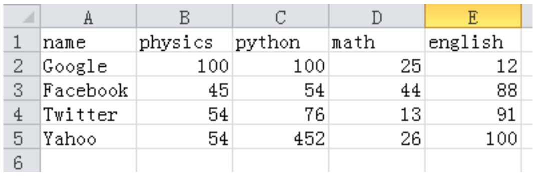
csv □□□ excel □□□□□□□□□□□ csv □□□□□□□

> □□□□□□Comma-Separated Values□CSV□□CSV □□

□□□□□□□□□□□□□□□□□□“□□□□□□”“□□□□□□□□□□□□”□□□□□□□□□□□□□□□□□□□□□□□□□□□ marks.csv □□□□□□□□□□□□□□□□□□□□□□□

```
name,physics,python,math,english
Google,100,100,25,12
Facebook,45,54,44,88
Twitter,54,76,13,91
Yahoo,54,452,26,100
```

□□□□□□□□□□□□□□□□□□□□□□□□□□



	A	B	C	D	E
1	name	physics	python	math	english
2	Google	100	100	25	12
3	Facebook	45	54	44	88
4	Twitter	54	76	13	91
5	Yahoo	54	452	26	100
6					

□□□□□□

□□□□□□□□□□ open() □□□□□

```
>>> with open("./marks.csv") as f:
...     for line in f:
...         print line
...
name,physics,python,math,english

Google,100,100,25,12
```

```
Facebook,45,54,44,88

Twitter,54,76,13,91

Yahoo,54,452,26,100
```

□□□□□□□□□□□□

Python □□□□□ csv □□□□□□□□ csv □□□□□□□□□

```
>>> import csv
>>> dir(csv)
['Dialect', 'DictReader', 'DictWriter', 'Error', 'QUOTE ALL',
'QUOTE MINIMAL', 'QUOTE NONE', 'QUOTE NONNUMERIC', 'Sniffer', 'StringIO',
' Dialect', '  all  ', '  builtins  ', '  doc  ', '  file  ', '  name  ',
'  package  ', '  version  ', 'excel', 'excel tab', 'field_size_limit',
'get dialect', 'list dialects', 're', 'reader', 'reduce',
'register_dialect', 'unregister_dialect', 'writer']
```

□□□□□□□□□□□□□□□□□□□□□□□□□□□□csv □□□□□□□□□□□□□□□□□□□□□□□□

```
>>> import csv
>>> csv reader = csv.reader(open("./marks.csv"))
>>> for row in csv_reader:
...     print row
...
['name', 'physics', 'python', 'math', 'english']
['Google', '100', '100', '25', '12']
['Facebook', '45', '54', '44', '88']
['Twitter', '54', '76', '13', '91']
['Yahoo', '54', '452', '26', '100']
```

□□□□□□□

□ Pandas □□

□□□□□□□□□□□□□□□□□□□□□ Pandas □□□□

```
>>> import pandas as pd
>>> marks = pd.read_csv("./marks.csv")
>>> marks
       name  physics  python  math  english
0    Google      100     100    25       12
1  Facebook       45      54    44       88
2   Twitter       54      76    13       91
3     Yahoo       54     452    26      100
```

□□□□□□□□□□□□□□□□□□□□□□□ Pandas □□□□□□□□□□□□□□□□□□□□□□ DataFrame □□□

□□□□□□□□□

```
>>> pd.read table("./marks.csv", sep=",")
       name  physics  python  math  english
0    Google      100     100    25       12
```

```
1  Facebook       45      54    44       88
2   Twitter       54      76    13       91
3     Yahoo       54     452    26      100
```

□□□□□□□□□□□□□□□□□ DataFrame □□□□□□□□□

```
>>> dir(marks)
['T', '_AXIS_ALIASES', '_AXIS_NAMES', '_AXIS_NUMBERS', '__add__',
'__and__', '__array__', '__array_wrap__', '__class__', '__contains__',
'__delattr__', '__delitem__', '__dict__', '__div__', '__doc__', '__eq__',
'__floordiv__', '__format__', '__ge__', '__getattr__', '__getattribute__',
'__getitem__', '__getstate__', '__gt__', '__hash__', '__init__',
'__iter__', '__le__', '__len__', '__lt__', '__module__', '__mul__',
'__ne__', '__neg__', '__new__', '__nonzero__', '__or__', '__pow__',
'__radd__', '__rdiv__', '__reduce__', '__reduce_ex__', '__repr__',
'__rfloordiv__', '__rmul__', '__rpow__', '__rsub__', '__rtruediv__',
'__setattr__', '__setitem__', '__setstate__', '__sizeof__', '__str__',
'__sub__', '__subclasshook__', '__truediv__', '__weakref__', '__xor__',
'_agg_by_level', '_align_frame', '_align_series', '_apply_broadcast',
'_apply_raw', '_apply_standard', '_auto_consolidate', '_bar_plot',
'_boolean_set', '_box_item_values', '_clear_item_cache', '_combine_const',
'_combine_frame', '_combine_match_columns', '_combine_match_index',
'_combine_series', '_combine_series_infer', '_compare_frame',
'_consolidate_inplace', '_constructor', '_count_level', '_cov_helper',
'_data', '_default_stat_axis', '_expand_axes', '_from_axes',
'_get_agg_axis', '_get_axis', '_get_axis_name', '_get_axis_number',
'_get_item_cache', '_get_numeric_data', '_getitem_array',
'_getitem_multilevel', '_helper_csvexcel', '_het_axis', '_indexed_same',
'_init_dict', '_init_mgr', '_init_ndarray', '_is_mixed_type',
'_item_cache', '_ix', '_join_compat', '_reduce', '_reindex_axis',
'_reindex_columns', '_reindex_index', '_reindex_with_indexers',
'_rename_columns_inplace', '_rename_index_inplace', '_sanitize_column',
'_series', '_set_axis', '_set_item', '_set_item_multiple',
'_shift_indexer', '_slice', '_unpickle_frame_compat',
'_unpickle_matrix_compat', '_verbose_info', '_wrap_array', 'abs', 'add',
'add_prefix', 'add_suffix', 'align', 'append', 'apply', 'applymap',
'as_matrix', 'asfreq', 'astype', 'axes', 'boxplot', 'clip', 'clip_lower',
'clip_upper', 'columns', 'combine', 'combineAdd', 'combineMult',
'combine_first', 'consolidate', 'convert_objects', 'copy', 'corr',
'corrwith', 'count', 'cov', 'cummax', 'cummin', 'cumprod', 'cumsum',
'delevel', 'describe', 'diff', 'div', 'dot', 'drop', 'drop_duplicates',
'dropna', 'dtypes', 'duplicated', 'fillna', 'filter', 'first_valid_index',
'from_csv', 'from_dict', 'from_items', 'from_records', 'get',
'get_dtype_counts', 'get_value', 'groupby', 'head', 'hist', 'icol',
'idxmax', 'idxmin', 'iget_value', 'index', 'info', 'insert', 'irow',
'iteritems', 'iterkv', 'iterrows', 'ix', 'join', 'last_valid_index',
'load', 'lookup', 'mad', 'max', 'mean', 'median', 'merge', 'min', 'mul',
'ndim', 'pivot', 'pivot_table', 'plot', 'pop', 'prod', 'product',
'quantile', 'radd', 'rank', 'rdiv', 'reindex', 'reindex_axis',
'reindex_like', 'rename', 'rename_axis', 'reorder_levels', 'reset_index',
'rmul', 'rsub', 'save', 'select', 'set_index', 'set_value', 'shape',
'shift', 'skew', 'sort', 'sort_index', 'sortlevel', 'stack', 'std', 'sub',
'sum', 'swaplevel', 'tail', 'take', 'to_csv', 'to_dict', 'to_excel',
'to_html', 'to_panel', 'to_records', 'to_sparse', 'to_string', 'to_wide',
'transpose', 'truncate', 'unstack', 'values', 'var', 'xs']
```

[illegible]

```
>>> marks.index
Int64Index([0, 1, 2, 3], dtype=int64)
>>> marks.columns
Index([name, physics, python, math, english], dtype=object)
>>> marks['name'][1]
'Facebook'
```

[illegible] DataFrame [illegible]

```
>>> marks.sort(column="python")
       name  physics  python  math  english
1  Facebook       45      54    44       88
2   Twitter       54      76    13       91
0    Google      100     100    25       12
3     Yahoo       54     452    26      100
```

[illegible]"Python"[illegible] Python [illegible]

```
>>> marks[:1]
     name  physics  python  math  english
0  Google      100     100    25       12
>>> marks[1:2]
       name  physics  python  math  english
1  Facebook       45      54    44       88
>>> marks["physics"]
0    100
1     45
2     54
3     54
Name: physics
```

[illegible] dir() [illegible] help() [illegible] pandas [illegible] pandas[illegible]

[illegible]

csv [illegible] MS excel [illegible] json [illegible] xml [illegible] pandas [illegible]

.xls [illegible] .xlsx

[illegible] excel [illegible]

```
>>> dir(pd)
['DataFrame', 'DataMatrix', 'DateOffset', 'DateRange', 'ExcelFile',
'ExcelWriter', 'Factor', 'HDFStore', 'Index', 'Int64Index', 'MultiIndex',
'Panel', 'Series', 'SparseArray', 'SparseDataFrame', 'SparseList',
'SparsePanel', 'SparseSeries', 'SparseTimeSeries', 'TimeSeries',
'WidePanel', '  builtins  ', '  doc  ', '  docformat  ', '  file  ',
'  name  ', '  package  ', '  path  ', '  version  ', ' engines',
' sparse', ' tseries', 'concat', 'core', 'crosstab', 'datetime',
'datetools', 'debug', 'ewma', 'ewmcorr', 'ewmcov', 'ewmstd', 'ewmvar',
'ewmvol', 'fama macbeth', 'groupby', 'info', 'io', 'isnull', 'lib',
'load', 'merge', 'notnull', 'np', 'ols', 'pivot', 'pivot table',
'read clipboard', 'read csv', 'read table', 'reset printoptions',
'rolling apply', 'rolling corr', 'rolling corr pairwise', 'rolling_count',
'rolling cov', 'rolling kurt', 'rolling max', 'rolling mean',
'rolling median', 'rolling min', 'rolling quantile', 'rolling_skew',
'rolling std', 'rolling sum', 'rolling var', 'save',
'set eng float format', 'set printoptions', 'sparse', 'stats', 'tools',
'util', 'value_range', 'version']
```

[illegible] `read_csv()` [illegible] `read_xls()` [illegible] `ExcelFile` [illegible]

```
>>> xls = pd.ExcelFile("./marks.xlsx")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/lib/pymodules/python2.7/pandas/io/parsers.py", line 575, in
  init
    from openpyxl import load workbook
ImportError: No module named openpyxl
```

[illegible] [illegible]pip [illegible] openpyxl [illegible]`sudo pip install openpyxl`[illegible]

```
>>> xls = pd.ExcelFile("./marks.xlsx")
>>> dir(xls)
['  class  ', '  delattr  ', '  dict  ', '  doc  ', '  format  ',
'  getattribute  ', '  hash  ', '  init  ', '  module  ', '  new  ',
'  reduce  ', '  reduce ex  ', '  repr  ', '  setattr  ', '  sizeof  ',
'  str  ', '  subclasshook  ', '  weakref  ', ' parse_xls', '_parse_xlsx',
'book', 'parse', 'path', 'sheet_names', 'use_xlsx']
>>> xls.sheet names
['Sheet1', 'Sheet2', 'Sheet3']
>>> sheet1 = xls.parse("Sheet1")
>>> sheet1
   0    1    2   3    4
0  5  100  100  25   12
1  6   45   54  44   88
2  7   54   76  13   91
3  8   54  452  26  100
```

[illegible]columns [illegible] csv [illegible]sheet1 [illegible] DataFrame [illegible]

[illegible] DataFrame [illegible]——[illegible] dir() [illegible] help() [illegible]——[illegible]

[illegible] json [illegible] xml [illegible](7)[illegible] Series [illegible] DataFrame [illegible]

[illegible] MySQL [illegible] (1) [illegible] Series [illegible] DataFrame [illegible]

[illegible]

[illegible] qiwsir@126.com, [illegible]

□□□□□

□□
□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ pandas □□□□□□□□□□□□□□

□□ yahoo □□□□

□□□□□□□□□□□ yahoo □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ yahoo□□□□□□□□□□□□□□□ www.yahoo.com□□□□□□□□□□□□□□□□□□

□□ yahoo □□□□□□□□□□□□□□□□□□□□□□□□□□□□ yahoo □□□□□□□□□□

```
 In [1]: import pandas
In [2]: import pandas.io.data

In [3]: sym = "BABA"

In [4]: finace = pandas.io.data.DataReader(sym, "yahoo",
start="2014/11/11")
In [5]: print finace.tail(3)
                 Open       High        Low      Close    Volume  Adj
Close
Date
2015-06-17  86.580002  87.800003  86.480003  86.800003  10206100
86.800003
2015-06-18  86.970001  87.589996  86.320000  86.750000  11652600
86.750000
2015-06-19  86.510002  86.599998  85.169998  85.739998  10207100
85.739998
```

□□□□□□□□□□□□ 2014 □ 11 □ 11 □□□□□□□□□□□□□

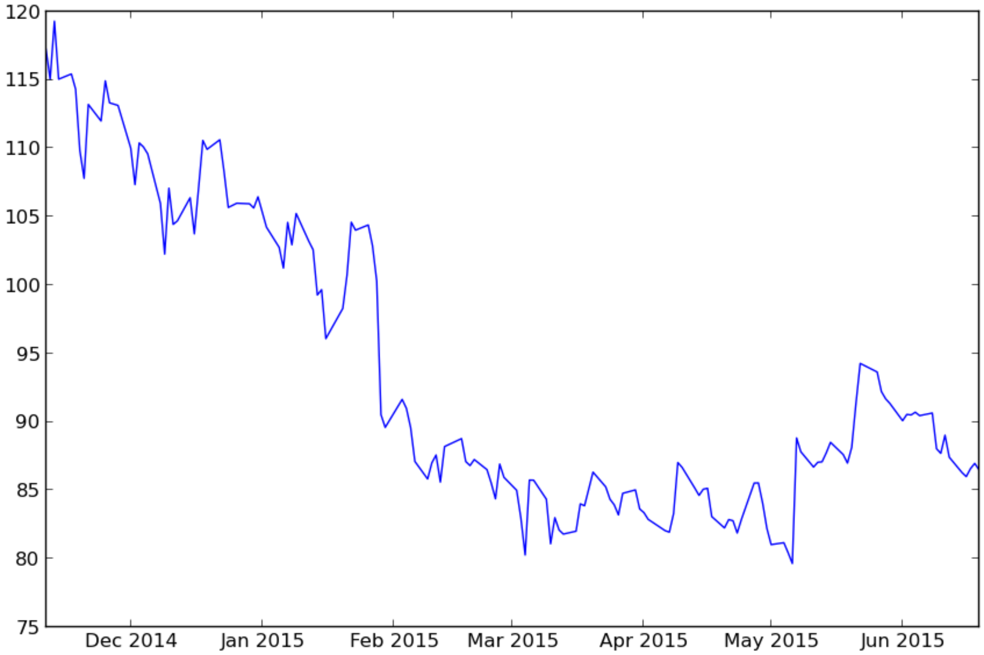
□□

□□□□□□□ DataFrame □□□□□□□□□□□□ finace □□□□□□□

```
 In[6]: import matplotlib.pyplot as plt
In [7]: plt.plot(finace.index, finace["Open"])
Out[]: [<matplotlib.lines.Line2D at 0xa88e5cc>]
```

```
In [8]: plt.show()
```

□□□□□□□□□



□□□□□□□□□□□□□□□□ 2014 □ 11 □ 11 □□ 2015 □ 6 □ 19 □□□□□□□□□□□□□□□□□“□□□”□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□ 3 □ 3 □□4 □ 4 □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□

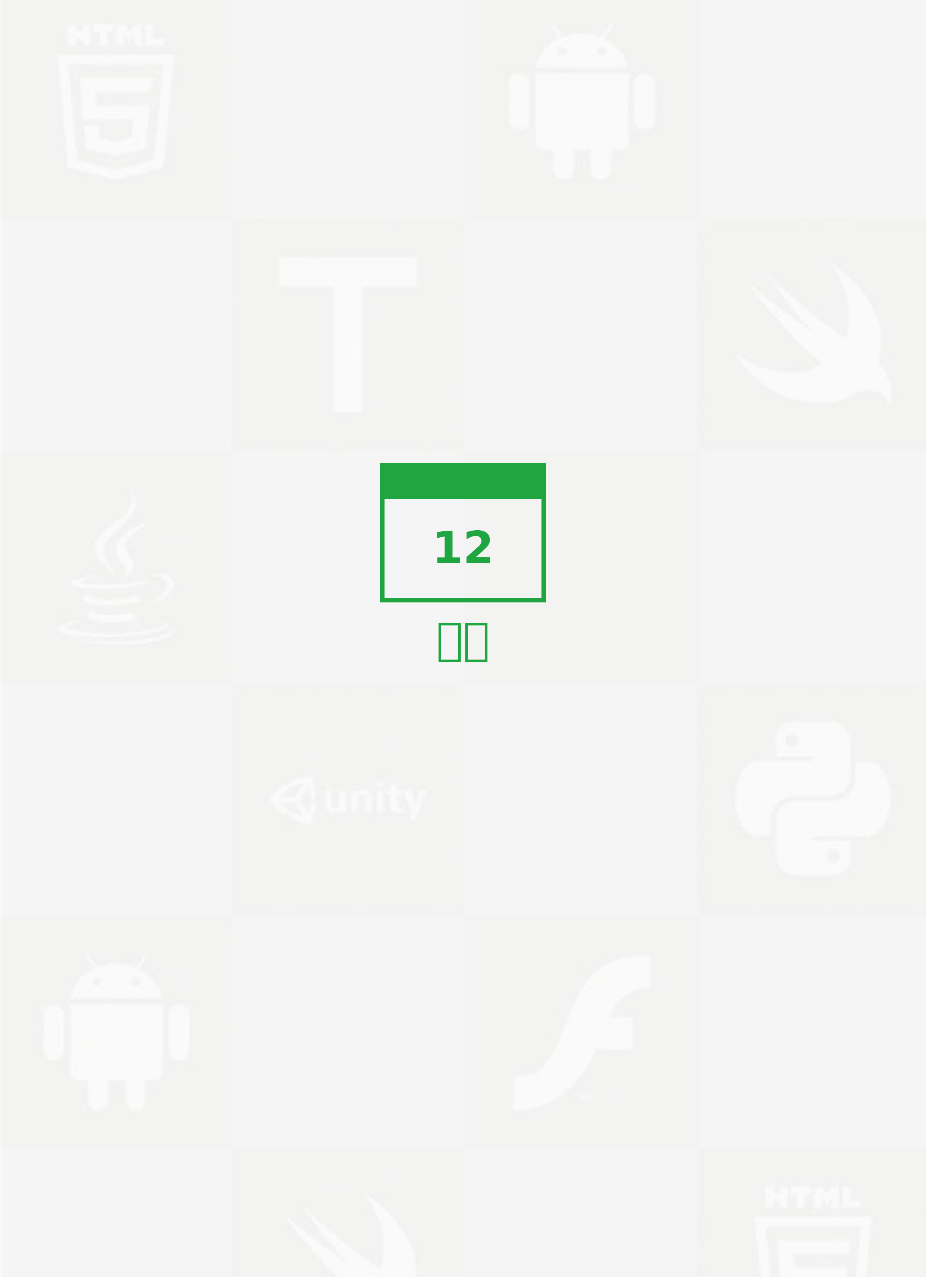
□□□□□□ `import matplotlib.pyplot as plt` □□□□□□□□□□`matplotlib` □□□ Python □□□□□ □□□“□□□”□□□□□□□□□□□ matplotlib □□□ □□□□□□matplotlib □□□□——John Hunter □□□ 2012 □ 8 □ 28 □□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□matplotlib.org□□□□□□□□□□□□□□

□□□□□□□□□□□□□ `dir()` □□□□□□□□□□□□□ finace □□□□ DataFrame □□□□□□□□□□□□□□□□□□□□□□ □□□□——`dir+help`——□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

[illegible] | [illegible]Pandas [illegible] (2)

[illegible]qiwsir@126.com,[illegible]



12

[illegible]

[illegible] Python [illegible]

[illegible] Python [illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]
- [illegible]

[illegible]/[illegible]

[illegible]

[illegible]([illegible]) —— [illegible]

[illegible] —— [illegible]

- http://www.amk.ca/python/writing/functional
- http://www.secnetix.de/olli/Python/lambda_functions.hawk
- http://docs.python.org/howto/functional.html

[illegible]

[illegible]“[illegible]”(Python[illegible]Ruby)[illegible]

[illegible] Python [illegible] Python [illegible] Python [illegible] —— [illegible] Youtube([illegible] Python [illegible])

- http://jaynes.colorado.edu/PythonIdioms.html
- http://wiki.python.org/moin/PythonSpeed/PerformanceTips

[illegible]

[illegible] TDD ([illegible])[illegible] BDD ([illegible])[illegible] TDD/BDD[illegible]

[illegible] —— [illegible]

- http://www.oreillynet.com/lpt/a/5463
- http://www.oreillynet.com/lpt/a/5584
- http://wiki.cacr.caltech.edu/danse/index.php/Unit_testing_and_Integration_testing
- http://docs.python.org/library/unittest.html

[illegible]

[illegible] —— [illegible]([illegible])[illegible]

[illegible]

- http://www.python.org/dev/peps/pep-0008/
- http://www.fantascienza.net/leonardo/ar/python_best_practices.html

[illegible] Python [illegible]

[illegible]

> [illegible]http://blogread.cn/it/article/3892?f=wb

极客学院
jikexueyuan.com
中国最大的IT职业在线教育平台

[illegible] ☞
http://wiki.jikexueyuan.com/project/start-learning-python/